U.S. Department of Transportation
Federal Aviation Administration
Office of Systems Engineering Management
Washington, D.C. 20591

Automated En Route Air Traffic Control Algorithmic Specifications

A 136796

SECTOR WORKLOAD PROBE

Volume 4

DTIC FILE COPY

JAN 1 3 1984

H

September 1983

Report No. DOT/FAA/ES-83/7

This document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161

84 01 13 083

Technical Report Documentation Page

1. Report No. DOT/FAA/ES-83/7	2. Government Accession No. A136796	3. Recipient's Catalog No.
4. Title and Subtitle Automated En Route Air Traffic Control Algorithmic Specifications SECTOR WORKLOAD PROBE Volume 4		5. Report Date September 1983
		6. Performing Organization Code FAA-ES-83-7
7. Author's) W.P. Niedringhaus, A.H. Gisch		8. Performing Organization Report No. AES-320
9. Performing Organization Name and Address Systems Engineering Service Department of Transportation Federal Aviation Administration 800 Independence Ave., S.W., Washington, D.C. 20591		10. Work Unit No. (TRAIS)
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address Same as #9 above.		13. Type of Report and Period Covered
		14. Sponsoring Agency Code AES
15. Supplementary Notes		

16. Abstract

This Algorithmic Specification establishes the design criteria for four advanced automation software package of the Advanced Automation System (AAS). The need for each function is discussed within the context of the existing National Airspace System (NAS). A top-down definition of each function is provided with descriptions on increasingly more detailed levels. The final, most detailed description of each function identifies the data flows and transformations taking place within each function.

This document consists of five volumes. Volume 4, Sector Workload Probe: Contains a functional design for the use of trajectory data and conflict data (both airspace and aircraft) to predict workload parameters which may assist supervisory air traffic control personnel in determining sector staffing levels and sectorization schemes in Air Route Traffic Control Centers.

The other four volumes of this specification provide design criteria for the following:

- Volume 1, Trajectory Estimation
- Volume 2, Airspace Probe
- Volume 3, Flight Plan Conflict Probe
- Volume 5, Data Specification

17. Key Words Automation, Air Traffic Control, Automated Decision Making, En Route Traffic Control, Artificial Intelligence, Advanced Automation System	18. Distribution Statement Document is available to the U.S. public through the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified		

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

EXECUTIVE SUMMARY

This specification establishes design criteria for a Sector Workload Probe algorithm, which is part of the initial automation for the Advanced Automation System of the Federal Aviation Administration's Air Traffic Control (ATC) System. This algorithm calculates measures related to workload. The algorithm takes into account a variety of measures. These measures include the following:

- average aircraft count;
- number of expected aircraft or airspace conflicts (as generated by two other advanced automation algorithms, Flight Plan Conflict Probe and Airspace Probe);
- a measure of actions which must be carried out by controllers;
- a density measure; and
- an overall measure.

For every sector, each measure is projected for various time intervals of approximately 15 minutes up to about two hours in the future.

An Area Supervisor or Area Manager may, at any time, request a display of the current and projected workload measures for a specified sector or set of sectors. Also, the Area Supervisor may monitor selected sector(s) to determine if certain measures exceed or fall below thresholds that he or she specifies.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Concluded)

LIST OF ILLUSTRATIONS

1. INTRODUCTION

The Federal Aviation Administration (FAA) is currently in the process of developing a new computer system, called the Advanced Automation System (AAS), to help control the nation's air traffic. The AAS will consist of new or enhanced hardware (i.e., Central Processing Units, memories, and terminals) and new software.

The new software will retain most or all of the functions in the existing National Airspace System (NAS) En Route Stage A software. The algorithms will need to be recoded and, in some cases, revised. In addition, the new AAS software will contain several new functions that make greater use of the capabilities of automation for Air Traffic Control (ATC). When fully implemented, these new functions are intended to detect and resolve many routine ATC problems.

The initial implementation of the AAS, described in the AAS Specification [1], will provide the ability to detect some common ATC problems. To meet the requirements of the AAS, several new ATC functions need to be postulated and described. Four of these functions are described in this document: Trajectory Estimation, Flight Plan Conflict Probe, Airspace Probe, and Sector Workload Probe [Volumes 1, 2, 3, and 4]. Together, they represent an initial level of automation and the beginnings of the evolution of the ATC system in accordance with the NAS Plan [2]. The NAS Plan presents an overview of the complete set of changes proposed to NAS in the coming decade.

1.1 Purpose

The purpose of this volume is to identify design criteria for Sector Workload Probe (SWP). SWP is one of the advanced automation functions called for in the AAS Specification. These design criteria specified in this volume are based on the existing National Airspace System (NAS) and the specification of the AAS. The AAS specification describes the Sector Workload Probe function and proposes some high level requirements for this function.

1.2 Scope

This algorithmic specification presents design criteria for a computational framework of Sector Workload Probe. The framework is a set of algorithms which collectively describe how it may be possible to measure the workload associated with the sectors which provide air traffic control services. It may

be viewed as a candidate for consideration in the final design. However, it is not intended to be the complete final design of SWP in the AAS.

The framework establishes the requirements for input and output data and provides a description of the flow of control of data as it is transferred from input to output. Some of the principal requirements have been identified in the "Operational and Functional Description of AERA 1.01" [3]. To the extent possible, the data are discussed using existing NAS terminology.

1.3 Organization of This Document

The remainder of Section 1 provides a description of Sector Workload Probe's role in the larger ATC context and in future enhancements of the ATC system. Both the operational considerations and processing methods of SWP are summarized. Section 2 defines the terminology used in the specification and discusses the factors which influence the design of the algorithms.

Descriptions of the algorithms are contained in Section 3, Sector Workload Probe Functional Design, and in Section 4, Detailed Description. The Sector Workload Probe Function, like the other advanced automation functions, is divided hierarchically into <u>subfunctions</u>, <u>components</u> and <u>elements</u> (underlined words in Sections 1 and 2 are critical to the understanding of this specification and can be found in the Glossary, Appendix B). Section 3 specifies the design, environment, and assumptions of the subfunctions (e.g., the Subject Aircraft Workload), and outlines their components (e.g., Individual Aircraft Workload Update). Section 4 provides a detailed description of each subfunction's components, including their mission, data requirements, and some processing details, and in some cases includes a discussion of a component's elements (e.g., Create Subject Sector Time Intervals).

Appendix A defines the data shared by the various subfunctions of SWP. (Similarly, Volume 5 of this document contains the global data shared by the functions defined in Volumes 1 through 4). Appendix B, as mentioned above, contains a glossary of those terms that are critical to an understanding of this specification.

A Program Design Language (PDL) which describes high level control logic using structured English is used as needed to describe the algorithms in this specification. A description of this PDL is contained in Appendix C. Finally, Appendix D provides a complete list of references.

1.4 Role of Sector Workload Probe in the Overall ATC System

This section discusses some of the features of the current ATC System, describes the role of Sector Workload Probe in the Advanced Automation System, and indicates changes to SWP that may be appropriate when enhancements to the AAS are introduced.

The Sector Workload Probe calculates measures related to workload, which is defined as any task performed by personnel to provide air traffic control services to aircraft.

1.4.1 System Context

The continental United States airspace is partitioned into 20 centers or Air Route Traffic Control Centers (ARTCCs), which control divisions of airspace bounded horizontally by polygons and stretching vertically from the center floor to 60,000 ft. Each center's airspace is further divided into areas, which in turn are divided into sectors. Areas and sectors are polygonal regions with floors (either at specified altitudes or the ground) and ceilings. The number of sectors within an area may change over time due to traffic volume changes and other factors. Each sector at any time is composed of one or more indivisible parts, which are called basic sectors in SWP. The process of combining or decombining basic sectors is called sectorization. A combined sector consists of one or more basic sectors under a sectorization plan.

Each sector is staffed by one or more radar controllers, whose responsibilities include maintaining aircraft separation standards and delivering maneuver clearances to aircraft. The Area Supervisor is the first line supervisor of radar controllers in an area. An Area Supervisor's responsibilities include the assignment of controllers to operating positions among the sectors in an area. The Area Manager is the second line supervisor of the radar controllers and is responsible for determining when to perform a sectorization. The Area Manager and/or an Area Supervisor also determine(s) whether to alter the number of controllers assigned to an area.

Hereafter, the term supervisor will designate either an Area Supervisor or an Area Manager. The term controller shall include radar controllers, but exclude supervisors.

The plans of the Federal Aviation Administration for the evolution of Air Traffic Control are discussed in "Advanced Automation System, System Level Specification" [1], and in "National Airspace System Plan (NASP): Facilities, Equipment

and Associated Development" [2]. According to the NASP, the "early capabilities [of Automated Air Traffic Control] will include . . . workload probe, which calculates predicted sector workload for use by [a] supervisor for determining sector staffing levels and sectorization for workload balancing." According to the AAS, the probe should "analyze the workload for each sector whenever a flight plan is activated or amended, or an active flight plan is received from an adjacent facility. This automated ATC function shall provide information to determine whether an unacceptable workload will be imposed on any control sector." The interpretation of these statements used in this specification is that sector workload is computed according to various measures; the supervisor may request to be notified when one or more measures meets conditions specified by him, or the supervisor may simply request SWP information at any time.

A workload measure or measure is a mathematical function which attempts to assign numerical values to one or more aspects of workload. Sector Workload Probe produces a set of workload measures using data from other functions of the Advanced Automation System as input. Together, the measures account for as many aspects of workload as possible, subject to the limitations of the data base.

1.4.1.1 Sector Workload Probe and AERA

The advanced automation functions for the ATC System described in this document and in the other algorithmic specifications [Vols. 1,2,3], are part of an automated system referred to as AERA ("Automated En Route Air Traffic Control"). AERA is to be implemented in several stages, as outlined in "Evolution of Advanced ATC Automation Functions" [4]. Sector Workload Probe will be implemented as part of the first stage, known as AERA 1 (subdivided into AERA 1.01 and AERA 1.02). Operational descriptions of the advanced automation functions of AERA 1 are given in "Operational and Functional Description of AERA 1.01" [3].

The Sector Workload Probe accesses various parts of the AERA data base as input. The sources for these data are: Trajectory Estimation, Flight Plan Conflict Probe (FPCP), and Airspace Probe (AP). No functions in AERA 1 depend upon the outputs of the Sector Workload Probe. However, later stages of AERA may use outputs of SWP. The Conflict Resolution function, for example, may use SWP output in the selection of maneuvers for aircraft.

SWP is unique among the AERA functions in that its output is presented to the supervisor. Other AERA functions either output no information or output information to the (radar) controllers.

1.4.1.2 En Route Sector Loading

The Sector Workload Probe may interface with a Central Flow Control (CFC) function, En Route Sector Loading (ELOD), which may receive input data from the SWP. ELOD, one of the major CFC enhancements planned for 1983 or early 1984 implementation, is claimed to improve the CFC processes of demand forecasting and prediction of en route saturation; these have historically been major weaknesses of CFC. ELOD determines areas of traffic saturation and provides "information" to flow management in the centers on a predictive rather than reactive basis. Its traffic demand projections are based on Official Airline Guide and General Aviation schedules, flight plans, manually entered data, and arrival times. The flight plan or arrival times may be updated under certain conditions. An alert message is generated if the projected traffic demand count for any sector or fix (specified point on the aircraft's route) during a designated time-interval (e.g., 15 minutes) exceeds an adapted threshold parameter.

Sector Workload Probe differs from En Route Sector Loading in two major respects:

- SWP has access to the full automation data base, in particular, the path of the aircraft through space and time (updated by radar reports), information from FPCP and AP, and density information. ELOD's primary input is flight plan information.

- SWP is a tool for the supervisor and does not directly impact the flight of any aircraft. When it indicates heavy workload, such as large traffic volume, the supervisor's response may be to sectorize or alter staffing—that is, change the airspace/ATC configuration but not the traffic flow pattern. When ELOD indicates heavy traffic, no resolution is created by CFC; however, the local flow management may change the traffic flow pattern but not the airspace/ATC configuration.

While the second of the above differences is fundamental, the first difference is due primarily to ELOD's early

implementation. ELOD could be enhanced by using the data created by SWP in the automation data base.

1.4.2 Role of Sector Workload Probe in Future System Enhancements

The role of SWP will undergo modifications in future enhancements of AERA as discussed below.

1.4.2.1 New Parameters and Changes in Parameters

As various ATC operations are automated, the associated SWP parameters will be determined and incorporated in the workload measures. Ongoing study, both before and after implementation of AERA, may provide improved understanding of the user-system interface issues involved, leading to refinements in the SWP measures and their associated parameters. In later stages of AERA implementation, when clearances are automatically delivered to the aircraft, the associated parameters will be redefined. As new types of planned actions are introduced, parameters may be introduced to measure their contribution to workload.

1.4.2.2 Long Range Probe

The Long Range Probe function will be incorporated into AERA 1 to help a controller decide whether to accept a proposed flight plan or flight plan amendment (e.g., for a user preferred route). Long Range Probe may be an extension of Flight Plan Conflict Probe and/or Airspace Probe. The algorithm and operational use of the Long Range Probe have not yet been determined.

The density workload measure of SWP is different in concept from the Long Range Probe (which was originally called Density Probe). The two functions serve different users (supervisors and controllers, respectively). However, some of Sector Workload Probe's measures, including the average aircraft count, the overall workload measure, and the density measure, may prove valuable sources of data for use by the Long Range Probe.

1.5 Sector Workload Probe Summary

This section describes SWP from an operational point of view and gives an overview of its internal functioning.

1.5.1 Operational Description

The supervisor may at any time request an immediate display of the current and projected workload for a specified sector or set of sectors. The supervisor may also request SWP to monitor selected sector(s) for certain conditions, such as a workload measure exceeding a selected threshold. Upon such a request, SWP periodically computes the measure(s) for the sector(s) and alerts the supervisor when these conditions are met.

The data displayed by SWP for the sector or sectors specified include average aircraft count, number of Flight Plan Conflict Probe and Airspace Probe encounters, and a measure of actions which are predicted to be carried out by the controller. Each measure is computed for various times in the near future. In addition, a measure of the density information and an overall measure may be displayed.

High SWP measures may influence the supervisor to increase staffing at a sector or to decombine a sector, while low measures suggest a decrease in staffing or the combining of two or more sectors into one.

1.5.2 Processing Overview

SWP processing is performed on data either for one aircraft or for all aircraft in a sector. The processing for one aircraft is performed when the aircraft's flight plan is first entered into the data base or is resynchronized or otherwise changed. Resynchronization is defined as the task of recomputing the estimated trajectory of an aircraft, when the trajectory is inconsistent with the aircrafts' recent history and the aircraft's last reported position. Trajectories are the modeled paths of aircraft through space and time.

Invocation of SWP occurs following automatic invocation of Flight Plan Conflict Probe and Airspace Probe. The SWP processing involves determining workload measures for the aircraft by sector for various times in the future. Much of SWP's input data is output directly from FPCP and AP. Other SWP information is determined directly from the new/resynchronized aircraft's trajectory.

If Sector Workload information is requested directly, the algorithm immediately computes the data to be displayed on all aircraft in a sector or group of sectors. If SWP is monitoring for a given condition, the algorithm computes and tests the data periodically.

2. DEFINITIONS AND DESIGN CONSIDERATIONS

Section 2 defines terms that will be used in the following sections and lists design considerations that affect the choice of an algorithm for SWP.

2.1 System Design Definitions

Some fundamental terms which are also used in other AERA functions have already been defined or discussed in Section 1. This section will define terms which are specific to SWP.

2.1.1 Time Terminology

The Sector Workload Probe evaluates its workload measures only up to a certain time bound, called the time horizon, which is a system parameter whose value is equal to the time horizon used by Flight Plan Conflict Probe. The common time horizon extends far enough into the future that most trajectories within a planning region are encompassed in their entireties (i.e., only a few extend beyond the time horizon). For similar reasons, it is advantageous to both FPCP and SWP to project trajectory information, as far into the future as possible, even though neither SWP nor FPCP has any outputs dependent on such projections. The justification for this is quite strong for FPCP [Vol. 3]; the justification in the case of SWP is largely based upon SWP's close interface with FPCP.

Should a trajectory contain a portion that extends beyond the time horizon, that portion is not processed immediately by FPCP or SWP. A list is kept of all such trajectories and is updated whenever the trajectories change. Periodically, at intervals of delta horizon, a determination is made whether a portion of any trajectory is now, with the passage of time, encompassed by the time horizon. These portions are treated like new trajectories, except that they are stored in the computer as extensions of the earlier portions. Each such occurrence for a new trajectory portion is called an SWP horizon update.

The SWP display time horizon is another system parameter specifying the maximum time in the future that SWP values are displayed. It precedes the time horizon, by perhaps one and a half to two hours. The period until the display time horizon is quantized into time-intervals of equal length. This length, another parameter, is called the time-interval duration and has a value of perhaps 15 minutes. Workload measures are calculated for these time-intervals (but not for any finer time quantization). The time periods associated with the time

horizon and display time horizon must be integer multiples of the delta horizon, which is, in turn, an integer multiple of the time-interval duration, itself a multiple of the cell width in the time dimension. Figure 2-1 illustrates the concepts developed in this section.

2.1.2 Sectorization Schedule

The sectorization plans consist of the possible ways that basic sectors can be combined via sectorizations. The sectorization schedule identifies the current sectorization plan as well as those sectorization plans (and their effective times) expected to be implemented over the interval between the current time and the time horizon.

2.1.3 SWP Trajectory Update

An SWP trajectory update is defined to include any of the following four events:

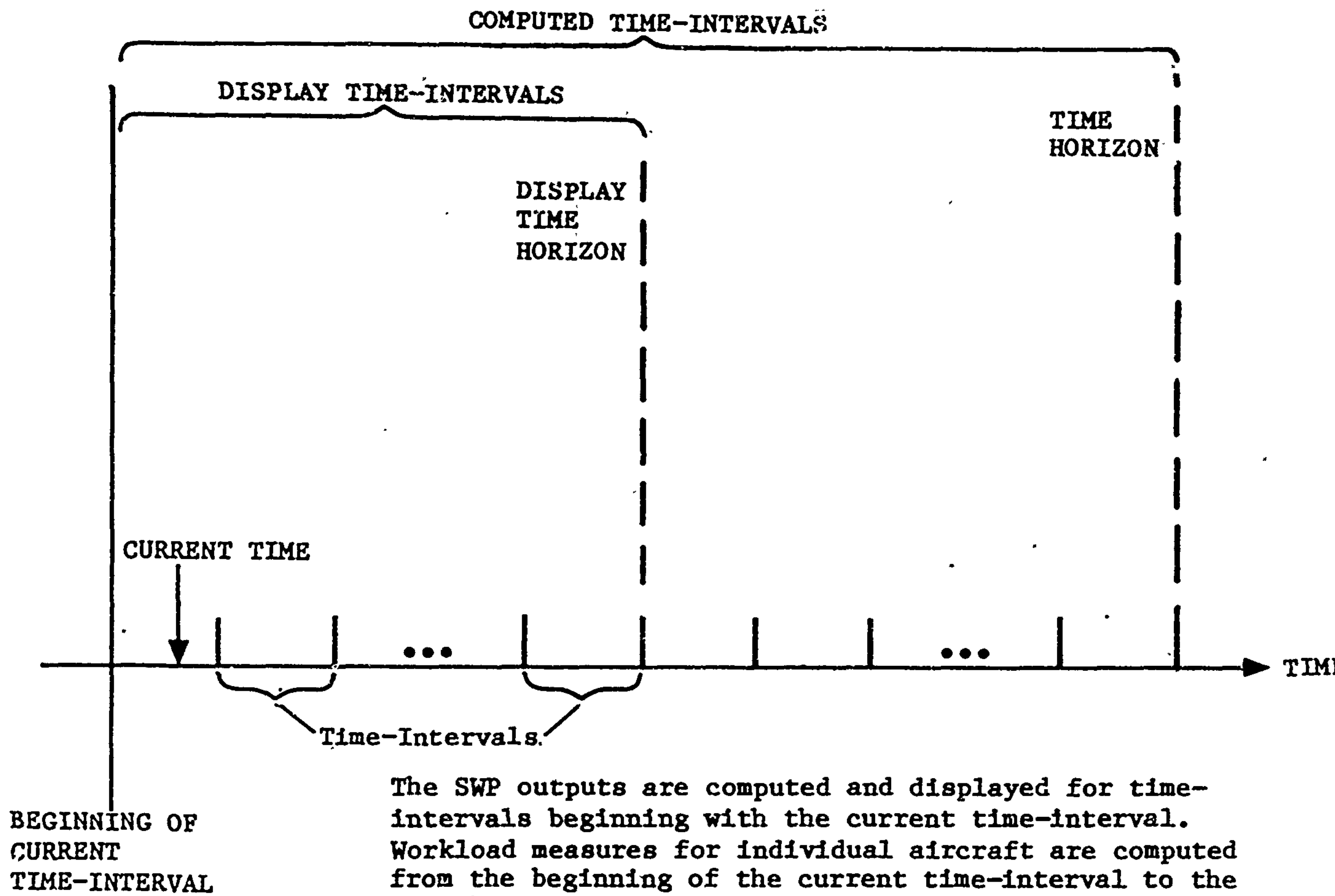
- A trajectory is added to the center's automation data base for the first time.
- A trajectory already in the center's automation data base is resynchronized by the Trajectory Estimation Function.
- A trajectory already in the automation data base is altered due to an action by a controller or by another AERA function.
- An SWP horizon update occurs.

Hereafter, the terms SWP trajectory update and SWP horizon update will be shortened to trajectory update and horizon update, respectively. (Note: FPCP uses slightly different definitions of these terms; the FPCP terms do not appear in this volume.)

Each trajectory update triggers a call to the Subject Aircraft Workload subfunction of Sector Workload Probe.

2.1.4 Subject and Object Aircraft

When an aircraft's trajectory is updated, the aircraft is designated as the subject. All other aircraft are called objects.



The SWP outputs are computed and displayed for time-intervals beginning with the current time-interval. Workload measures for individual aircraft are computed from the beginning of the current time-interval to the time horizon. Workload measures for the sector are computed and displayed until the display time horizon.

FIGURE 2-1
TIME TERMINOLOGY

2.1.5 Immediate Mode and Conditional Mode

When a supervisor requests to see an immediate display of sector workload information, SWP is said to be in Immediate Mode. The request is called an Immediate Mode Request. If a supervisor requests to see a display of sector workload information when conditions (specified by him or her) are satisfied, SWP is said to be in Conditional Mode. The request is then called a Conditional Mode Request.

2.1.6 The Airspace Grids

SWP uses several grids, which have various of the dimensions (x,y,z,t) to represent the airspace. The discrete compartments of each grid are called grid cells or cells. The grids and their dimensions are the following:

- The sector airspace grid to store sector boundaries (x,y,z)
- A grid to represent trajectories of aircraft, computed by Flight Plan Conflict Probe (x,y,t)
- The cell density airspace grid, to compute a fine measure of density (x,y,z,t)
- The block density airspace grid, to compute a coarse measure of density (x,y,z,t)

The first three grids share common widths in the x and y dimensions which are doubled in size in the fourth grid. Each grid's cells are squares in the (x,y) plan. All cell extents in the z dimension (first, third and fourth grid) are equal, but the common z extent may vary with altitude above the ground. Cell extents in the t dimension are shared by the second and third grid; this extent is doubled in the fourth.

The sector airspace grid assigns each of its cells to a single basic sector. It is adapted to the site, and is used only as input to SWP.

The second grid is computed by Flight Plan Conflict Probe [Vol. 3, Section 2.1]. The trajectory of an aircraft traverses a sequence of cells. A cell is said to be occupied if the trajectory satisfies certain FPCP and SWP criteria with respect to that cell. The set of cells occupied by an aircraft's trajectory is called the aircraft's grid chain. Holding patterns are also included in the grid chain. The grid chain computed by

FPCP works satisfactorily for SWP (assuming a common grid size as discussed in Section 2.2.6). There is no necessity for SWP to redo the FPCP calculations for generating a grid chain given an approximation of an aircraft's trajectory.

The third and fourth grids are used by SWP to accumulate counts of the number of aircraft occupying each grid cell for computation of the density measure. For convenience, when these two grids are discussed, the terms cell and block are used to refer to cells in the cell density airspace grid and the block density airspace airspace grid, respectively. Each block, then, contains eight cells. The sizes of the x, y, and t dimensions of a block are twice those of a cell, but the size of the z dimension of a block is the same as that for a cell.

The extent of cells (and blocks) in the z dimension (used in the first, third and fourth grids) may take different values with different altitudes above the ground. The z extent is small enough to allow all parts of each cell to be within or near enough to its basic sector that events within the cell contribute to workload for that basic sector. Values for the cell size in z above 6000 feet appear incompatible with this constraint. On the other hand, the values of cell size in z must be large enough to have a good probability of grouping aircraft together which are close enough in altitude over the time horizon interval, so that their interaction is included as a contribution to workload. The time horizon is far enough in the future that values of cell size in z below 3000 feet appear inconsistent with this constraint. Other factors that may affect the selection of cell size in the z dimension are the locations of flight level boundaries and the altitudes most commonly used to divide sectors along the vertical axis.

2.1.7 Nominees and Encounters

In Flight Plan Conflict Probe, an object aircraft is called a nominee if it and the subject aircraft (a) occupy the same grid cell or adjacent cells and (b) are separated vertically by less than a vertical separtion criterion. All nominees are tested for horizontal separation with the subject aircraft. If separation is predicted to fall below a certain specified threshold referred to in Vol. 3 as advisory Seph), then the nominee is called an encounter.

Analogously, in Airspace Probe, an object (in this case a polygon of airspace) is also called a nominee if it and the subject aircraft trajectory occupy the same grid cell or adjacent cells. The nominees are tested for horizontal separation with

certain restricted or warning airspaces in the planning region. If this separation falls below a threshold, the nominee is then called an encounter.

The numbers of FPCP and AP encounters are useful indicators of sector workload.

2.1.8 Planned Actions

A planned action for an aircraft is any one of a set of actions that can be anticipated. The workload of a planned action is expected to be performed for the aircraft at some time t, called the activation time (an input to SWP). Planned actions are output by various AERA functions. Activation times are computed from data associated with the planned actions. Planned actions are used internally in AERA 1; they are not, like encounters, intended to be warnings to the controller. In AERA 1 the planned actions consist of vectors, altitude changes, altitude changes with restrictions, speed changes,and holding patterns.

2.1.9 Workload

Workload is defined to be those tasks performed to provide air traffic control services. References to workload in this document are associated with sector workload as opposed to controller workload. Sector workload includes those tasks performed in a sector, while controller workload consists of the actual work performed by controllers at the control positions. Sector workload is easier to measure than controller workload because controller workload is dependent on human factors, i.e., variability in the types of ATC methods among controllers, user-system interactions and the number of controllers assigned to a sector.

Workload measures may be expressed as weighted sums of submeasures, which are defined as the finest subdivision of workload that is calculated. For example, the planned action measure has a submeasure for each type of planned action.

2.2 System Design Considerations

This section discusses the design considerations that must be taken into account while designing algorithms for SWP.

2.2.1 No Background Knowledge Required of User

The supervisor should be able to use the SWP measures without detailed knowledge of the algorithm that generates them.

A very high-level understanding of the algorithm may be useful, however. The supervisor may receive instruction on how to interact and utilize the features of the output options and the workload measures.

2.2.2 Display Considerations

This specification does not address the formats that may be used to display SWP data, nor the manner in which a supervisor may input a request for SWP activation. It is assumed that the display is flexible and easy to use (perhaps menu driven), has some standard editing capability, and can satisfy both the supervisor who wishes to explore every feature as well as the supervisor who wishes to minimize the time required to learn how the function is used.

2.2.3 Limitations of Sector Workload Probe

SWP makes no attempt to measure workload directly in terms of staff-hours. Such a measurement would be impossible and misleading to the supervisor, given the limited data available to AERA. Even with full (and time consuming) cooperation on the part of the controller, it is not possible to divide the process of decision-making into time-measured portions divided among various tasks.

SWP has no knowledge of current staffing levels or of individual capabilities of controllers. The SWP may be more useful to the supervisor if the supervisor sets the thresholds to reflect this information.

The Sector Workload Probe is not intended to be used by controllers (e.g., for determining whether an individual aircraft may be granted a change in flight plan). As a tool for the supervisor, SWP does not itself make suggestions concerning management matters (such as the amount of staffing), but it does provide information to support such decisions.

2.2.4 Workload Allocated by Place and Time

Each event (such as an encounter or a planned action) that contributes to a workload measure is allocated to a specific place and time: the basic sector in which it occurs and the time-interval (Section 2.1.1) in which it is contained.

2.2.5 Encounters, Planned Actions, Display Times, and Sector Workload

Consider an encounter pair for which the FPCP horizontal and vertical separation criteria are predicted to be first violated at time t. It is inappropriate to assign the workload involved in resolving the encounter at time t to the (basic) sector(s) the aircraft occupy at time t, since the role of FPCP is to alert the controller long enough in advance of a separation violation to allow him to resolve the conflict routinely. Therefore, the workload is assigned to the sector which receives the advisory message from FPCP and to the time this message is displayed. This time is called the encounter's display-as-advisory time, and is input to SWP for each encounter. Figure 2-2 shows the interrelation between this time and the time of violation of horizontal separation criterion. Figure 2-3 shows how the workload may be allocated to sectors based on encounter display-as-advisory times.

For an Airspace Probe encounter, the workload (although different from that for an FPCP encounter) is likewise assigned to the time-interval containing the display time and to the sector which receives the message of an AP encounter.

Similar timing considerations apply as well to the workload measure based on the planned actions. AERA accounts for controller (and pilot and aircraft) reaction time by modeling aircraft conformance to lag the activation time. The controller is expected to have thought about the planned action prior to the activation time and to communicate the clearance for the planned action following that time. It is possible to imagine cases where the controller's think time exceeds or is exceeded by his performance time. Such considerations go beyond the scope of this specification. The workload is simply assigned to the time interval containing the activation time and to the (basic) sector then occupied by the aircraft.

2.2.6 Interface Between FPCP and SWP

This specification assumes that SWP and FPCP use airspace grids with the same cell sizes in the x, y, and t dimensions. This assumption may allow savings in computer storage and execution time. The width of the grid cells in the horizontal and time dimensions are system parameters. There are certain implications in setting them equal for the two functions.

The exact horizontal dimensions of the grid cells are, within broad limits, not critical to SWP. They are critical to FPCP

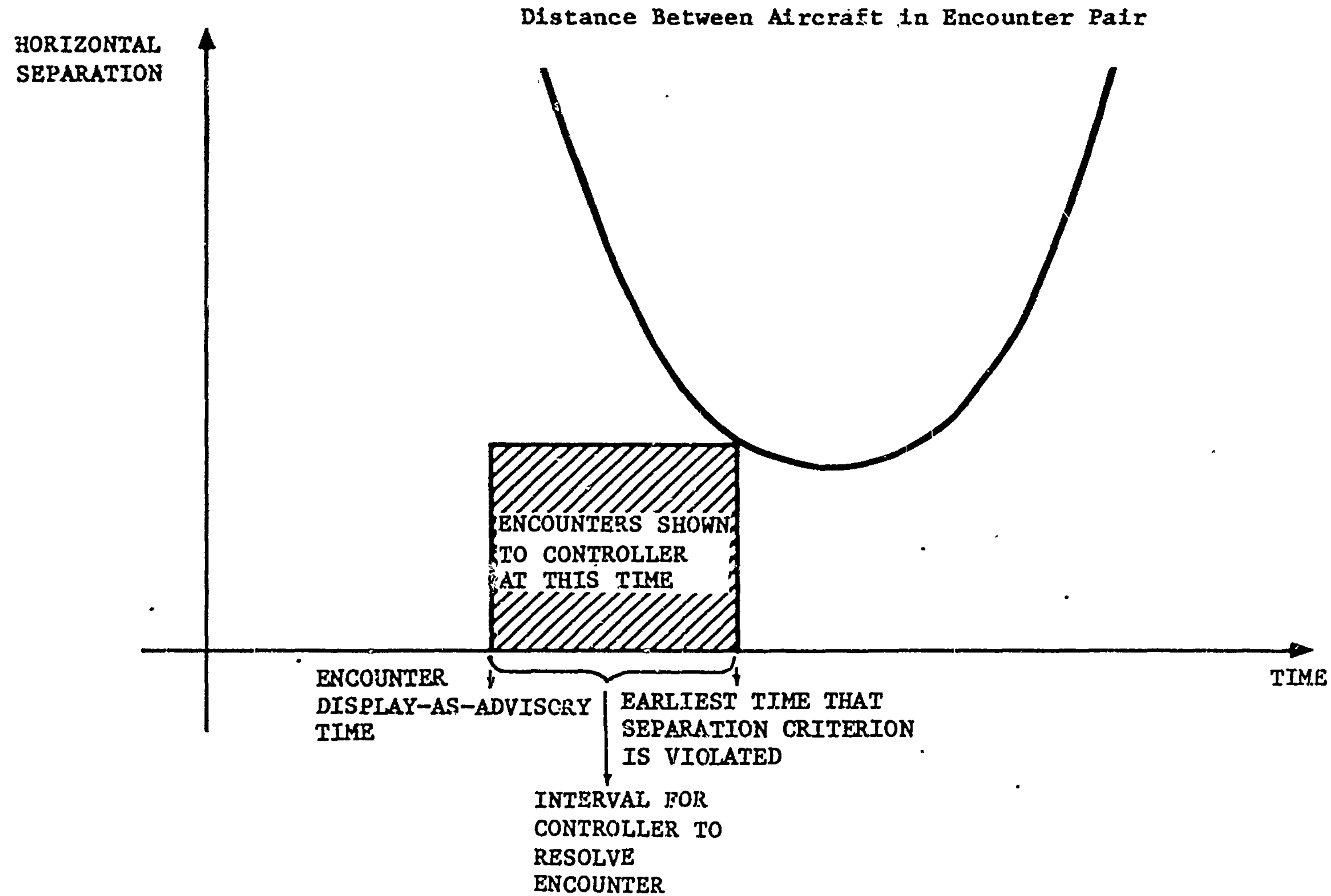
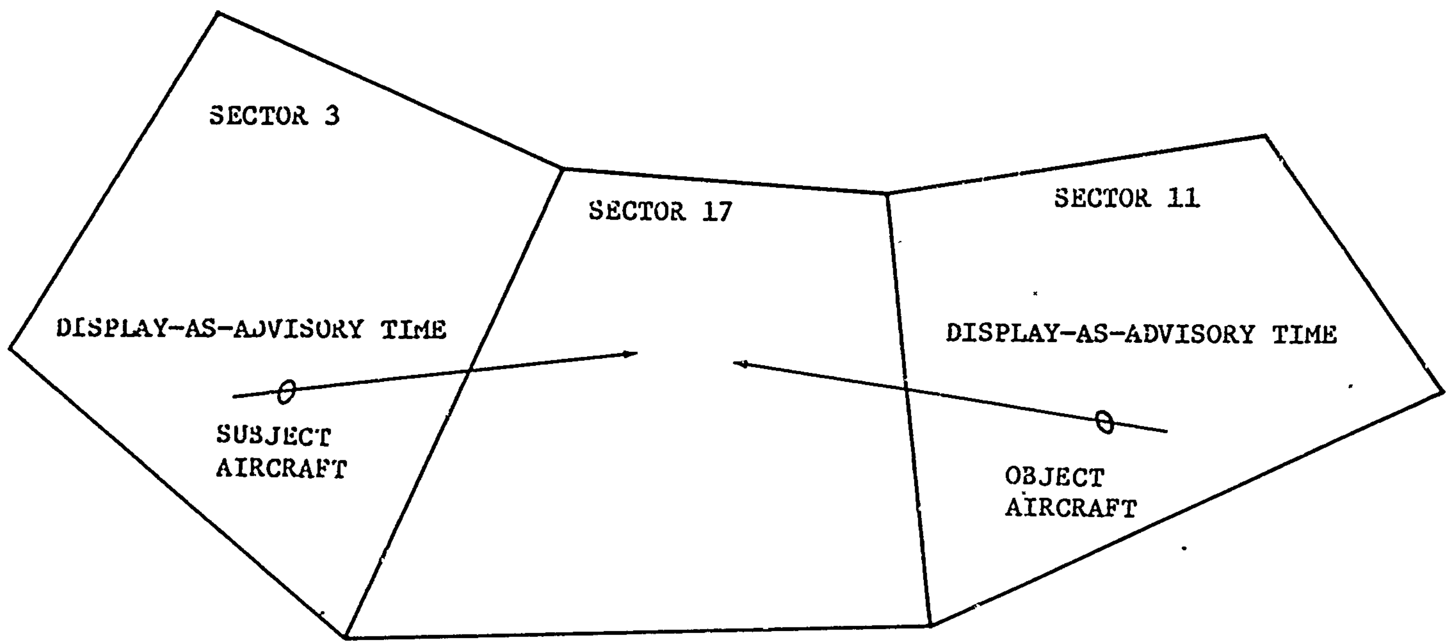


FIGURE 2-2
SEQUENCE OF EVENTS ASSOCIATED WITH A VIOLATION OF SEPARATION CRITERION



A conflict will occur in Sector 17, but the workload involved in resolving the conflict may be assigned (for the subject) to Sector 3 (where the conflict is displayed at display-as-advisory time) and (for the object) to Sector 11 (where the conflict is displayed simultaneously).

FIGURE 2-3
POSSIBLE ASSIGNMENT OF WORKLOAD FOR FPCP ENCOUNTER

and depend on the exact value of the advisory horizontal separation criterion [Vol. 3, Section 2.1.10] which is still to be determined but is expected to fall within SWP's broad limits. On the other hand, the exact dimensions of the grid cells in the time dimension are, within broad limits, not critical to FPCP but are critical to SWP: their time extent multiplied by 2^n must equal the time-interval duration, for some positive integer n. Should further study prove that the optimal range of the grid cell's time extent for FPCP does not contain SWP's time-interval duration divided by 2^n, grid-size-commonality implies that the cell's time extent must be rounded up to that value, at the cost of more nominees and more calls to the FPCP Fine Filter. There are certain advantages, however, even apart from SWP considerations, for the FPCP grid to be divided (such as SWP's time-intervals) on the time dimension in natural clock increments, even if these are not quite optimal in reducing nominees. For instance, for a trial FPCP probe, the controller might want a list of all encounters with display-as-advisory times earlier than 3:00 p.m.

The costs associated with assuming common grids for both SWP and FPCP appear to be outweighed by the benefits. It is worth noting, however, in view of the fact that SWP and FPCP system parameter values have yet to be determined, that different grid sizes could be used for SWP and FPCP without substantial changes in either algorithm. In this case, all references in this volume to the airspace grid cells created by FPCP would apply to corresponding data created by SWP. The FPCP Grid Chain Generator [Vol. 3, Section 4.1] would simply be run twice, once for each grid.

2.2.7 Access to Sector Workload Probe Information

SWP information may be obtained for an area or for a set of basic sectors. Supervisors may access data on other areas as well as their own.

Access to SWP information is not limited to the supervisor's own area. Data for any sector in the center are available for access. Information on workload in adjacent sectors may be useful in helping a supervisor evaluate factors including the following:

- Whether the supervisor's own controllers should alleviate their workload by routing aircraft through the adjacent sectors

- Whether there is a likelihood of an adjacent sector's personnel alleviating heavy workload by routing aircraft through the supervisor's area

- Whether coordination is necessary with adjacent flow control facilities or with Central Flow Control

It is not expected that controllers (nonsupervisors) will use SWP data, although no reasons have been identified to withhold this information should a controller request it.

2.2.8 Events at Sectorization

SWP is not explicitly intended to be used on a trial basis to determine what workload values would occur under a sectorization plan not currently in effect. The SWP thus differs in this respect from the other AERA 1 probes (Airspace Probe and FPCP), which are intended to allow trial probes of proposed flight plans. A supervisor should, however, as a part of a normal error-recovery process, be able to input and verify any sectorization plan before committing the system to it. The results of a trial sectorization plan (before commitment) should be available only to the supervisor initiating such a plan.

2.2.9 Adaptation of Workload Measures

Multiplicative constants called adaptation factors may be applied to the displayed workload measures to account for the decreasing quality and quantity of data and the increasing chance of data modification by resynchronization with increasing prediction times. The adaptation factors were created offline based on the historical data on the predicted versus the observed SWP workload measures and overall measures. The adaptation factors may depend on the sector, season of the year, day of the week, time of day, and how far ahead the prediction is made.

The adaptation factors are determined for each workload measure and for various values of v (number of time-intervals) and t (time at which SWP is invoked) from the following:

1. Predicted historical expected value of the workload measure at time t, projected v time-intervals into the future (e.g., the average workload calculated at 8 PM for 9 PM on Sundays in July for Sector 5)

2. Observed historical value of the workload measure v time-intervals after time t (e.g., the average workload calculated at 9 PM for 9 PM on Sundays in July for Sector 5)

3. Predicted current value of the workload measure at time t, projected v time-intervals into the future (e.g., the workload calculated at 8 PM for 9 PM on a particular Sunday in July for Sector 5)

The first two values are SWP constants (for various values of v, t, season, day of week, etc.) but the third value is calculated on-line. From these three values, the adaptation factor is calculated for that sector for a particular choice of v and t. For instance, as a rush hour builds, workload predictions based on trajectory data may be underestimates, causing the values of adaptation factors to be greater than one. As the rush hour diminishes, predictions may be overestimates, causing the values of the adaptation factors to be less than one.

3. SECTOR WORKLOAD PROBE FUNCTIONAL DESIGN

SWP is divided into two subfunctions: Subject Aircraft Workload and Supervisor Requests.

3.1 Environment

This section describes input data required by the SWP algorithm, conditions causing activation of the algorithm, and output data produced.

3.1.1 Input Data and Activation

3.1.1.1 Input Data

The input data for the Sector Workload Probe are global data and may be outlined as follows:

- Output from other AERA Functions
  - FPCP grid chain
  - AP encounter data
  - FPCP encounter data
    - Before trajectory update
    - After trajectory update
  - Planned action data
- Site-specific data
  - Sectorization plans
  - Sectorization schedule
  - Adaptation factors
  - Geographic extent of basic sectors (the sector airspace grid)
- Parameters
  - Subject aircraft identity
  - Time-interval duration
  - Time horizon
  - Display time horizon
  - Weighting coefficients
  - Frequency-of-testing parameter
  - AP encounter notification time parameter
- Supervisor requests
  - Sector(s) to display outputs
  - Time-interval(s) to evaluate
  - Condition(s) to test (specify sector number, workload measures and threshold values)
  - Display editing features

3.1.1.2 Automatic Activation Sequences

The Subject Aircraft Workload Subfunction is automatically activated by any trajectory update for an aircraft in the automation data base. During Conditional Mode, the Workload Evaluation Component of the Supervisor Requests Subfunction is automatically activated on a periodic basis.

3.1.1.3 Supervisor Initiating Sequences

The Supervisor Request Subfunction is initiated by the supervisor by an Immediate Mode Request, a Conditional Mode Request, or a request to edit the display.

3.1.2 Output Data

3.1.2.1 Output to the Global Data Base

SWP outputs to the global data base are used by the display function, but not by any other AERA 1.01 function. (In the future, outputs may be used by ELOD or other AERA functions as discussed in Section 1.4.1.1 and 1.4.1.2). The outputs are in the form of tables. The global data base receives two types of outputs from SWP, as described below. Both refer to aggregate aircraft data (i.e., data computed for each aircraft and then summed over all aircraft).

The first type of output consists of a table for the aggregate aircraft in each basic sector called the BASIC SECTOR WORKLOAD MEASURES (BSWM) Table. Table 3-1 presents one method of organization for the data. The BSWM Table is updated with each trajectory update. The table includes a record for each pair of basic sectors and time-intervals from the present to the time horizon.

The fields include the following:

- Total flight time
- Total FPCP encounter count
- Total AP encounter count
- Total count on planned actions for each type of planned action
- Average aircraft count
- Weighted sum of the planned actions
- Density measure
- Overall measure

TABLE 3-1
CONCEPTUAL STRUCTURE OF BASIC_SECTOR_WORKLOAD_MEASURES TABLE

Sector Number	Time Interval Id	Aircraft Minutes	FPCP Encounter Count	AP Encounter Count	Planned Action Count				
					Altitude Change	Altitude Change w/ Restrictions	Speed Change	Vector	Hold
1	1								
1	2								
1	:								
1	(last)								
2	1								
:	:								
(last)	1								
	(last)								

This is a possible format for the BASIC_SECTOR_WORKLOAD_MEASURES Table. Note that the average aircraft count, the density measure, the weighted sum of the planned actions, and the overall measures are not included in the table. These are computed just prior to the display of the outputs.

The second type of output consists of a table for the aggregate aircraft in each combined sector in an area. This table is known as the COMBINED SECTOR WORKLOAD MEASURES (CSWM) Table. The data for this table are computed when the workload is evaluated for the requested area. The COMBINED_SECTOR_WORKLOAD_MEASURES Table contains the same types of information as the BASIC_SECTOR_WORKLOAD_MEASURES Table, as well as three additional fields for the density measure.

3.1.2.2 Output to the Supervisor

The output to the supervisor consists of a subset of the following workload measures displayed for the organization type and time-intervals specified. If the organization type is an area, then the outputs are subsets of the COMBINED_SECTOR_WORKLOAD_MEASURES (CSWM) Table. If the organization type is sectors, then the outputs are subsets of the BASIC_SECTOR_WORKLOAD_MEASURES (BSWM) Table. These outputs include the:

- Average aircraft count
- Number of encounters for both (FPCP and AP)
- Weighted sum of the planned actions
- Density measure
- Overall measure (a weighted sum of the above outputs)

The number of each type of planned action may, upon supervisor request, be displayed. Also the values may be displayed for the measures adjusted by the adaptation factors.

3.2 Design Assumptions

The design of the SWP algorithm is based on the assumptions discussed below.

3.2.1 Considerations Regarding Conditional Mode and Immediate Mode

In Conditional Mode, the supervisor must specify each condition to be monitored: the sector(s) and workload measure(s) to test, and the threshold(s) to test against. AERA provides no default thresholds for Conditional Mode.

The supervisor may make Immediate Mode Requests during Conditional Mode without affecting Conditional Mode. The supervisor may cancel Conditional Mode or modify at any time the conditions specified.

3.2.2 Sectorization

Several design assumptions are made for the capability of verifying a new sectorization plan or a trial SWP. A trial SWP is initiated by the supervisor as an Immediate Mode Request. The outputs for a trial sectorization plan will be based on the combined sectors in the specified area instead of the basic sectors, which would not be affected by a change in the sectorization schedule. The main distinction between a trial SWP and a normal request for a workload evaluation of an area is that a different sectorization schedule is accessed (see Section 2.2.8).

3.3 Subfunctions of Sector Workload Probe

There are two subfunctions in the SWP algorithm: the Subject Aircraft Workload subfunction and the Supervisor Requests subfunction. This section gives an overview of these subfunctions and discusses changes that may be appropriate as later phases of AERA are introduced.

3.3.1 Subject Aircraft Workload

When a trajectory update occurs, the Subject Aircraft Workload Subfunction updates the workload data on the subject aircraft. Points on the trajectory where the aircraft enters either a new sector or a new time-interval are identified. These points subdivide the trajectory into portions called sector-time-intervals. All workload occurring within a sector-time-interval is treated as a unit. The values for each workload submeasure in each sector-time-interval are calculated and stored for the subject. The values for the aggregate aircraft (in the Basic Sector Workload Measures Table) are updated according to the net change due to the trajectory update.

3.3.2 Supervisor Requests

The Supervisor Requests Subfunction receives and processes various data entered by the supervisor. To evaluate and display the workload values for a specified sector (an Immediate Mode Request), the subfunction uses the data on aggregate aircraft. The workload submeasures in the BASIC_SECTOR_WORKLOAD_MEASURES Table are summed for each time-interval over each basic sector comprising the combined sector. The overall measure for each time-interval (weighted sum of the FPCP encounter count, the AP encounter count, the planned actions, and the density measures) is computed. The Supervisor Requests Subfunction also processes Conditional Mode Requests by the

supervisor and handles editing of display features for the table formats and other console options.

3.4 Expandability

As the development of AERA continues, as described in "Operational and Functional Description of the AERA Packages" [5], some enhancements will be necessary for the Sector Workload Probe. The values of the weighting coefficients for the automated activities (such as planned actions) may decrease as automation proceeds towards automatic decision making and clearance delivery. For those planned actions not implemented in AERA 1.01, the values of the weighting coefficients are included as the planned actions are implemented. Eventually, automation will play a predominant role in controlling routine traffic, with the controller monitoring the system and handling exception conditions. There may then be a need to reevaluate the Sector Workload Probe algorithm, since the factors affecting workload may be different.

Some expandability may occur in user-system interfaces for SWP. A supervisor, such as the Area Manager, may desire to view the outputs of several areas at the same time. The supervisors may specify the display features that they desire for initialization of the displayed output options when they are using SWP.

4. DETAILED DESCRIPTION

The Subject Aircraft Workload and Supervisor Requests subfunctions are activated under different circumstances. The activation of the first subfunction is automatic and is initiated by a trajectory update. A supervisor may activate the second by direct request. The Subject Aircraft Workload subfunction calculates an aircraft's contribution to workload according to the workload measures. The Supervisor Requests subfunction processes various requests from the supervisor and displays the outputs in the format he or she has defined.

4.1 Subject Aircraft Workload Subfunction

The Subject Aircraft Workload subfunction is invoked automatically with the occurrence of a trajectory update, following invocation of the Flight Plan Conflict Probe.

The values of several variables or parameters are required as inputs in order that the Subject Aircraft Workload subfunction can set the values of the corresponding shared local parameters:

- Subject flight identifier (Sflid)
- Status of the trajectory update (Trajectory_Update_Status)
- AP notification duration (AP_Not_Dur)

The AP_Not_Dur is the time duration between the time the controller is given a message of the AP encounter (immediately after the encounter is determined) and the time of penetration of the airspace by the aircraft.

Also, the initialization of several input tables is required prior to invocation of the Subject Aircraft Workload subfunction. These tables are the following:

- Sector airspace grid (SWP_CELL)
- Cell density airspace grid (CELL_DENSITY)
- Block density airspace grid (BLOCK_DENSITY)
- BASIC_SECTOR_WORKLOAD_MEASURES

Since the data in the sector airspace grid are created by center personnel and updated infrequently, the data are stored for access by SWP. When the automation data base is initialized, the cell density airspace grid and the block density airspace grid are created with zeroes for the values of aircraft count. Maintenance of these tables is required to remove those records in the tables for which the time values are no longer current

and to add records for future time values not in the table. These operations are not discussed in this volume.

When the automation data base is initialized, the BASIC_SECTOR_WORKLOAD_MEASURES Table is created with the basic sectors and the time intervals designated but with values of zero for the other fields. Maintenance of these records is also needed to update the time intervals. No outputs of this subfunction are directly intended for operational use. However, the outputs INDIVIDUAL_AIRCRAFT_WORKLOAD (IAW) Table, BASIC_SECTOR_WORKLOAD_MEASURES Table, cell density airspace grid data (CELL_DENSITY) and block density airspace grid data (BLOCK_DENSITY) are used by the Supervisor Requests subfunction to produce outputs for operational use.

Two components make up this subfunction and they are processed in the following order:

- Individual Aircraft Workload Update
- Basic Sector Workload Update

Figure 4-1 shows the structure of the components for the Subject Aircraft Workload subfunction.

4.1.1 Individual Aircraft Workload Update Component

4.1.1.1 Mission

For each aircraft, SWP maintains records in a table called the INDIVIDUAL AIRCRAFT WORKLOAD (IAW) Table. Table 4-1 illustrates some typical IAW data. The IAW table includes a record for each of an aircraft's sector-time-intervals, according to the aircraft's latest trajectory update. Each pair and its record corresponds to a sector-time-interval (Section 3.3.1).

The IAW table contains a field for each of the following:

- Sector number
- Time interval identifier
- Beginning time in the sector-time-interval
- Ending time in the sector-time-interval
- Total flight time
- FPCP encounter count
- AP encounter count
- Count of planned actions for each planned action type

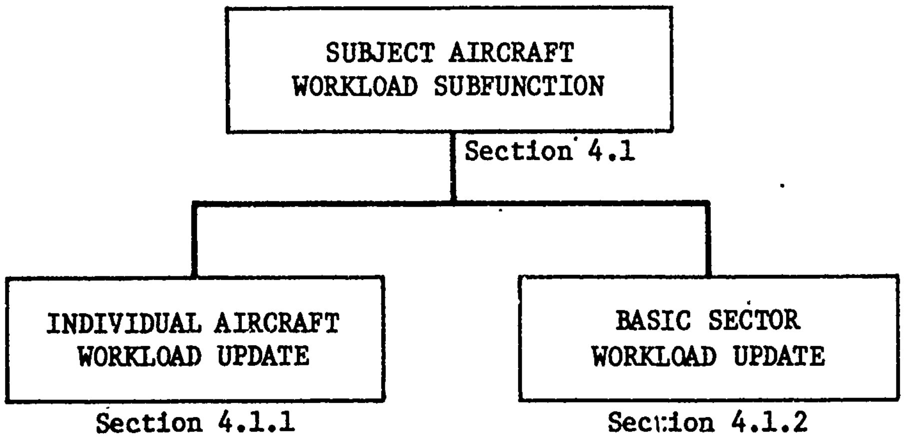


FIGURE 4-1
SUBJECT AIRCRAFT WORKLOAD SUBFUNCTION

TABLE 4-1
EXAMPLE OF INDIVIDUAL_AIRCRAFT_WORKLOAD TABLE

Flight Id 63				Aircraft Minutes	FPCP Encounter Count	AP Encounter Count	Planned Action Count				
Sector Number	Time Interval Id	Beginning Time	Ending Time				Altitude Change	Altitude Change w/ Restrictions	Speed Change	Vector	Hold
9	5	1:08	1:15	7	0	0	0	1	0	0	0
9	6	1:15	1:23	8	1	0	2	1	2	1	0
3	6	1:23	1:27	4	0	0	0	0	0	0	0
17	6	1:27	1:30	3	0	0	0	0	0	0	0
17	7	1:30	1:45	15	2	0	1	0	1	2	0
17	8	1:45	1:49	4	0	0	0	0	0	0	0

The Individual Aircraft Workload Update component creates a local table called the REVISED SUBJECT WORKLOAD (RSW) Table for the subject. The fields of this table are identical to those of the INDIVIDUAL_AIRCRAFT_WORKLOAD (IAW) Table. The RSW Table stores workload information for the subject according to the current trajectory update; the IAW Table contains, when this component is invoked, workload information for the subject according to its previous trajectory update, if any.

This component also updates the IAW Table for certain object aircraft--those for which the FPCP encounter information has changed due to the trajectory update. These aircraft are called encounter-status-changed or ESC aircraft. There are three reasons the encounter status may change:

- An encounter has become a non-encounter.
- A non-encounter has become an encounter.
- A previous encounter has remained, but its display-as-advisory time has changed, with a possible associated change in basic sectors.

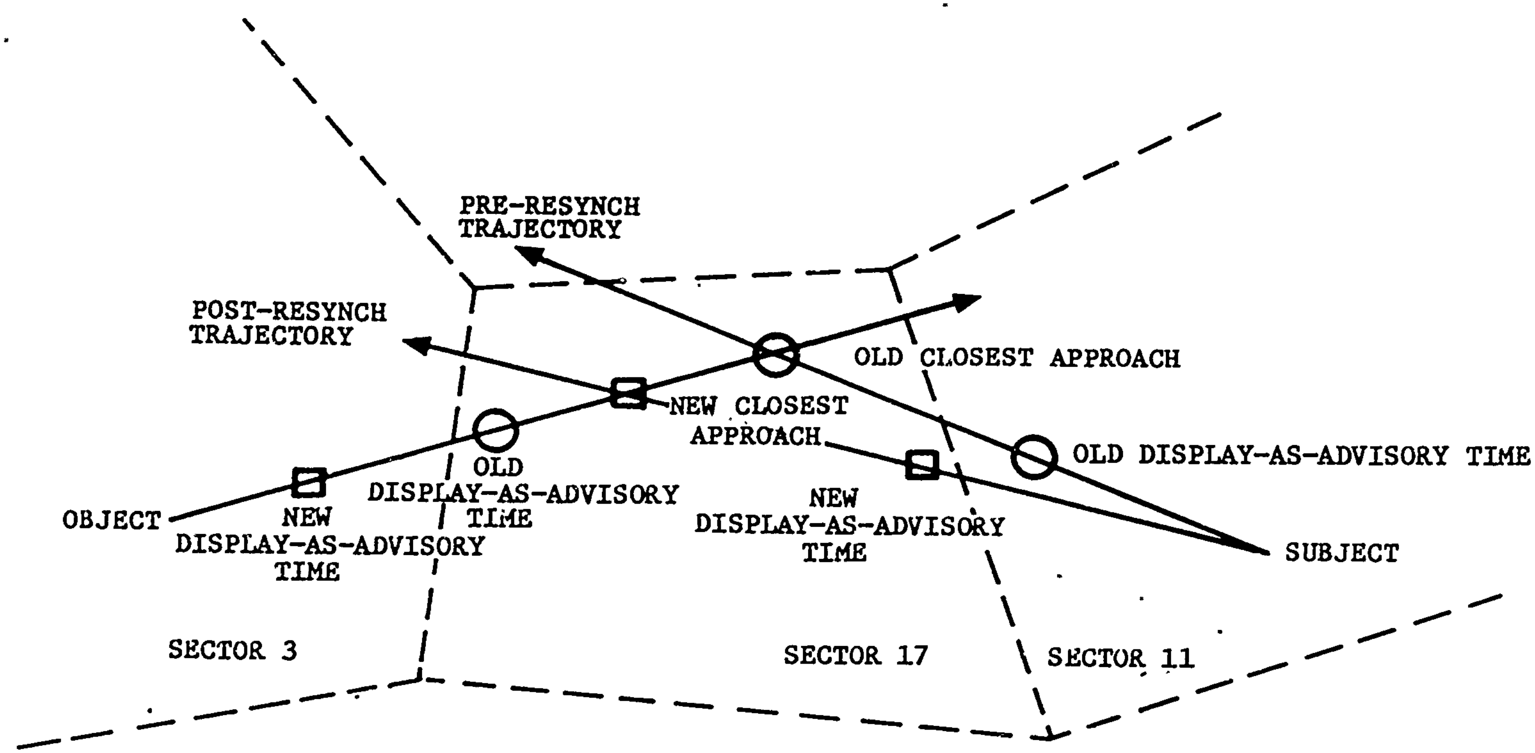
One example of the third type (as shown in Figure 4-2) is a resynchronization of the subject which changes the sector and time-interval of the ESC aircraft's display-as-advisory time.

4.1.1.2 Design Consideration and Component Environment

Input

The primary inputs are the trajectory update and the IAW Table. Other inputs include the following:

- SWP Parameters
  - Time-interval duration (Time_Interval_Duration)
  - Time horizon (Time_Horizon)
  - AP encounter notification time parameter (AP_Not_Dur)
- AERA Global Data Base
  - FPCP encounters (ENCOUNTERS and PRIOR_ENCOUNTERS)
  - AP encounters (ENVIRONMENTAL_CONFLICT)
  - Planned actions (PLANNED_ACTIONS and PLANNED_ACTION_DURATION)
  - Grid chain for subject (SPARSE_CELLS)



A resynchronization has moved the subject's display-as-advisory time so that its workload is credited to Sector 17 rather than 11. Likewise the object's workload is credited to Sector 3 rather than 17.

FIGURE 4-2
EFFECT OF RESYNCHRONIZATION ON ENCOUNTER DISPLAY-AS-ADVISORY TIMES AND LOCATIONS

- Sectorization schedule (SECTORIZATION_SCHEDULE)
- Sectorization plans (SECTORIZATION_PLAN)

- Subject Aircraft Identity (Sflid)

- Status of Trajectory Update (Trajectory_Update_Status)

- Cell density airspace grid data for all aircraft (CELL_DENSITY)

- Block density airspace grid data for all aircraft (BLOCK_DENSITY)

- Cell density airspace grid data by subject (SUBJECT_CELL)

- Block density airspace gird data by subject (SUBJECT_BLOCK)

Output

The outputs of the Individual Aircraft Workload Update component are the REVISED_SUBJECT_WORKLOAD (RSW) Table for the subject aircraft, the INDIVIDUAL_AIRCRAFT_WORKLOAD Table, and the ESC Tables. The ESC Tables contain the sector-time-intervals (with FPCP encounters) (a) created and (b) deleted from the IAW Table due to changes in encounter display-as-advisory times for the ESC objects identified by the trajectory update. The tables consist of the ESC_INCREMENT Table and the ESC_DECREMENT Table, respectively. The ESC Tables and the RSW Table are shared local and serve only as inputs to the next component (Basic Sector Workload Update). The RSW Table is eventually copied into the IAW Table.

For each sector-time-interval (record) in the IAW (and RSW) Table, the following information (fields) is recorded:

- Time-interval identifier
- Sector number
- Beginning time (in this sector-time-interval)
- Ending time (in this sector-time-interval)
- Total flight time (ending time minus beginning time)
- Number of AP encounters
- Number of FPCP encounters
- Number of each planned action type

4.1.1.3 Component Design Logic

The Individual Aircraft Workload Update component consists of the following elements which are called in sequence as outlined in Figure 4-3:

Create Subject Sector Time Interval
  Calculate Subject Density
  Calculate Sector
  Determine Unique Sector Time Intervals
Include Subject Encounters
Include Object Encounters
Include Subject Planned Actions

If the trajectory has been revised, then the former density data for the subject (on the cell and block levels) are removed from the airspace density grids for all aircraft. The aircraft count data for the subject in the airspace density grids are initialized to zero. First the sector-time-intervals are created and entered in the RSW Table. Then the encounters detected by FPCP and AP are counted in the appropriate sector-time-intervals of the RSW Table. The FPCP encounters are likewise included in the appropriate records of the IAW Table for the corresponding objects. If the subject's trajectory has been previously processed by SWP, then the previous FPCP encounters for the subject are removed from the IAW Table and are identified in the ESC_DECREMENT Table for further processing. Finally, the counts on the various planned actions are added to the RSW Table for the relevant sector-time-intervals.

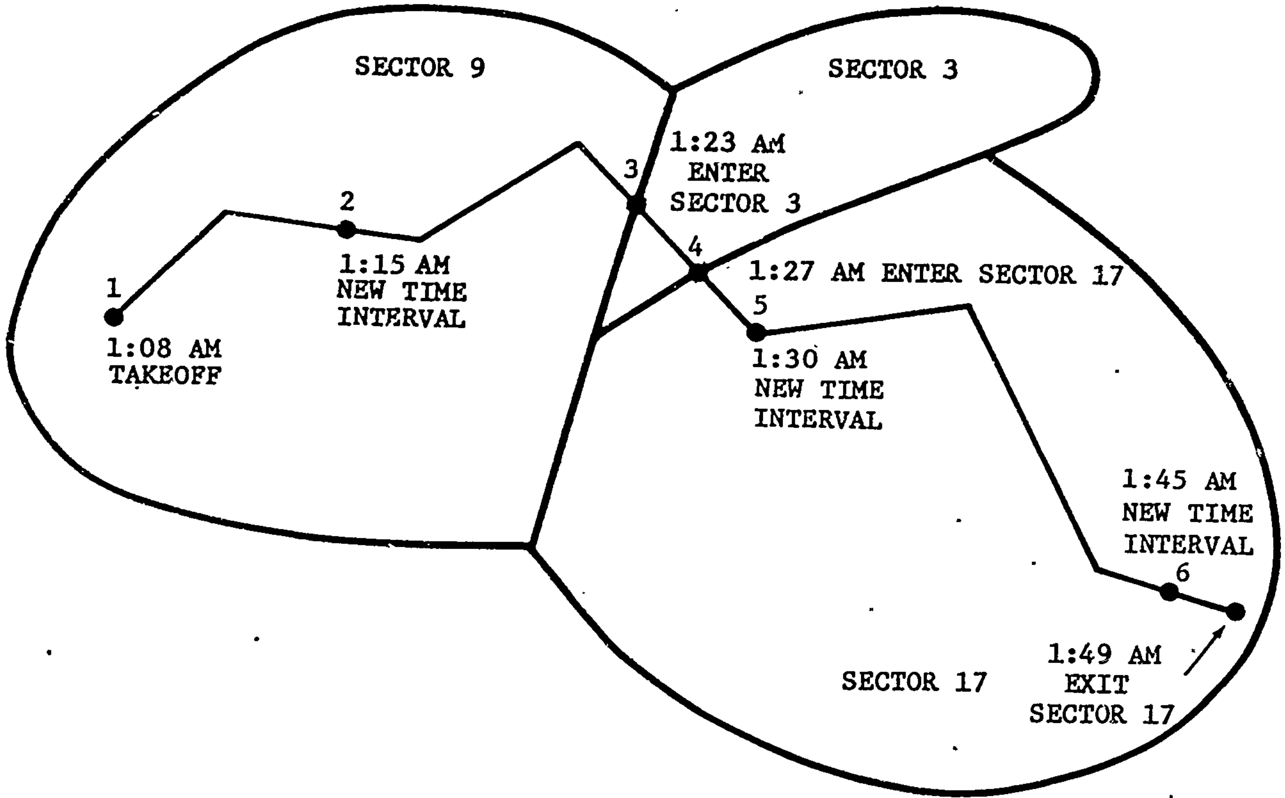
Create Subject Sector Time Intervals

In example of how the sector-time-intervals are determined is illustrated in Figure 4-4. The aircraft begins in Sector 9 and travels through Sector 3 and Sector 17. A new sector-time-interval corresponds to another record in the RSW Table. Table 4-1 represents a RSW Table for the aircraft depicted in Figure 4-4. In this example, fifteen minutes is used for the value of the time-interval duration and the time-interval id is set to the number of time-intervals elapsed since noon.

In the Create Subject Sector Time Interval element, information on each cell in the subject's grid chain is used to identify the sector-time-intervals and to determine the subject's contribution to the sector density. Each sector-time-interval is identified by unique beginning and ending times. The Create Subject Sector Time Interval element, illustrated in Figure 4-5, first calculates the impact of the subject on sector

```
ROUTINE Individual_Aircraft_Workload_Update;
REFER TO SHARED LOCAL Sflid IN, Trajectory_Update_Status IN,
    SUBJECT_CELL INOUT, SUBJECT_BLOCK INOUT, CELL_DENSITY INOUT,
    BLOCK_DENSITY INOUT;
    IF Trajectory_Update_Status EQ 'REVISED'
    THEN #remove former density data for subject#
       REPEAT FOR EACH SUBJECT_CELL RECORD
          WHERE SUBJECT_CELL.fl_id EQ Sflid AND SUBJECT_CELL.
             aircraft_count GT 0;
          UPDATE IN CELL_DENSITY (aircraft_count = aircraft_count - 1)
             WHERE CELL_DENSITY.cell_id EQ SUBJECT_CELL.cell_id AND
                CELL_DENSITY.beg_time EQ SUBJECT_CELL.beg_time;
          UPDATE IN SUBJECT_CELL (aircraft_count = 0);
       REPEAT FOR EACH SUBJECT_BLOCK RECORD
          WHERE SUBJECT_BLOCK.fl_id EQ Sflid AND SUBJECT_BLOCK.
             aircraft_count GT 0;
          UPDATE IN BLOCK_DENSITY (aircraft_count = aircraft_count-1)
             WHERE BLOCK_DENSITY.block_id EQ SUBJECT_BLOCK.block_id
                AND BLOCK_DENSITY.beg_time EQ SUBJECT_BLOCK.beg_time;
          UPDATE IN SUBJECT_BLOCK (aircraft_count = 0);
    ELSE #initialize density data for subject#
       REPEAT FOR EACH CELL_DENSITY RECORD;
          INSERT INTO SUBJECT_CELL (fl_id = Sflid, cell_id = CELL_
             DENSITY.cell_id, beg_time = CELL_DENSITY.beg_time,
             aircraft_count = 0, block_id = CELL_DENSITY.block_id);
       REPEAT FOR EACH BLOCK_DENSITY RECORD;
          INSERT INTO SUBJECT_BLOCK (fl_id = Sflid, block_id = BLOCK_
             DENSITY.block_id, beg_time = BLOCK_DENSITY.beg_time,
             aircraft_count = 0);
    CALL Create_Subject_Sector_Time_Interval;
    CALL Include_Subject_Encounters;
    CALL Include_Object_Encounters;
    CALL Include_Subject_Planned_Actions;
END Individual_Aircraft_Workload_Update;
```

FIGURE 4-3
INDIVIDUAL_AIRCRAFT_WORKLOAD_UPDATE



The trajectory of a subject aircraft is depicted for its horizontal (xy) profile. The beginning point of each sector-time-interval is identified by a number. The reason for each sector-time-interval is noted as either an entry to a sector or an entry to a time-interval.

FIGURE 4-4
ASSIGNING SECTOR-TIME-INTERVALS

```
ROUTINE Create_Subject_Sector_Time_Interval;
REFER TO GLOBAL SPARSE_CELLS IN, Time_Interval_Duration IN;
REFER TO SHARED LOCAL Sflid IN, Delta_T IN, REVISED_SUBJECT_WORKLOAD
   OUT;
#RSW is not saved after the logic is completed for the subfunction#
#Subject Aircraft Workload Update which uses this element#
DEFINE VARIABLES
   No_End_Time          Number records in End_Time Array
   Time_Interval_Id     Time interval being processed
   Basic_Sector_Number  Basic sector being processed
   Start_Cell_Time      Beginning time for x,y,t cell
   End_Cell_Time        Ending time for x,y,t cell
   No_Sec               Number of unique sectors in RSW with the same
                           ending time
   No_Same_Time         Number of records in RSW with same ending time
   End_Time (*)         Array of unique ending times in RSW
   I                    Index;
```

FIGURE 4-5
CREATE_SUBJECT_SECTOR_TIME_INTERVAL

```
CALL Calculate_Subject_Density;
REPEAT FOR EACH SPARSE_CELLS RECORD
   WHERE SPARSE_CELLS.fl_id EQ Sflid;
   CALL Calculate_Sector (SPARSE_CELLS IN, Basic_Sector_Number OUT);
   #one record in SPARSE_CELLS#
   Calculate Start_Cell_Time and End_Cell_Time from time part of
      SPARSE_CELLS.tree_node_id and Time_Interval_Duration;
   Time_Interval_Id = time interval with Start_Cell_Time and End_
      Cell_Time;
   INSERT INTO REVISED_SUBJECT_WORKLOAD
      # 1 record in RSW corresponds to 1 in SPARSE_CELLS#
      (sector_number = Basic_Sector_Number, time_interval_id = Time_
      Interval_Id, beginning_time = Start_Cell_Time, ending_time =
      End_Cell_Time, total_fl_time = 0, pa_counts = 0);
SELECT FIELDS UNIQUE ending_time
   FROM REVISED_SUBJECT_WORKLOAD
   RETURN COUNT (No_End_Time)
   INTO End_Time    #array#
   ORDER BY ending_time;
FOR I = 1 TO No_End_Time; #loop on number of unique ending times#
   SELECT FIELDS ending_time
      FROM REVISED_SUBJECT_WORKLOAD (RSW)
      RETURN COUNT (No_Same_Time)
      WHERE RSW.ending_time EQ End_Time(I);
   IF No_Same_Time GT 1
   THEN #decide if aircraft in more than 1 sector with this cell's#
      #ending time#
      SELECT FIELDS UNIQUE sector_number
         FROM REVISED_SUBJECT_WORKLOAD (RSW)
         WHERE RSW.ending_time EQ End_Time(I)
         RETURN COUNT (No_Sec);
         #No_Same_Time is still more than 1#
      IF No_Sec GT 1
      THEN #decide which sector to use for this end time#
         SELECT FIELDS sector_number
            FROM REVISED_SUBJECT_WORKLOAD (RSW)
            INTO Basic_Sector_Number
            WHERE RSW.ending_time EQ End_Time(I) - Delta_T;
         DELETE FROM REVISED_SUBJECT_WORKLOAD (RSW)
            WHERE RSW.ending_time EQ End_Time(I)
               AND RSW.sector_number NE Basic_Sector_Number;
CALL Determine_Unique_Sector_Time_Intervals;
      #to make in RSW one record per sector time interval#
END Create_Subject_Sector_Time_Interval;
```

FIGURE 4-5
CREATE_SUBJECT_SECTOR_TIME_INTERVAL (Concluded)

density on both the cell and block level as discussed in Figure 4-6. The sector and some time information are computed for each cell. The REVISED_SUBJECT_WORKLOAD Table undergoes several stages from the case where each record contains information on each cell in the subject's grid chain to the case where each record represents a unique sector-timeinterval. In the first stage of the RSW Table, each cell in the subject's grid chain (SPARSE_CELLS) is examined. The sector which contains the cell is determined by the Calculate Sector element shown in Figure 4-7. The Start_Cell_Time, End_Cell_Time, and Time_Interval_Id are computed and these are included in a record for the cell in the RSW Table. The Start_Cell_Time is the earliest time associated with the specific cell in the grid chain, while the End_Cell_Time is the latest time associated with this cell. The Time_Interval_Id designates the time interval which encompasses the time range of this cell. In the second stage of the RSW Table, the unique ending times in RSW are stored and processed individually. The REVISED_SUBJECT_WORKLOAD Table may contain multiple records with the same ending time (i.e., the x and y values changed with the t values remaining the same). If such multiple records exist for the same ending time, then tests are performed to determine which sector to associate with these RSW records. If the RSW Table contains more than one record with the specific cell ending time, then the routine determines if the aircraft is in more than one sector among these records. When these records have more than one sector with the same cell ending time, the sector with the last processed cell ending time is chosen as the sector with the current cell ending time and the records with other sectors for the current cell ending time are deleted. After all unique ending times have been processed, the Determine Unique Sector-Time-Intervals element produces the stage of the RSW Table where each record represents a sector-time-interval. In this element, depicted in Figure 4-8, the unique sector and time-interval pairs are selected and their beginning and ending times are computed to produce the RSW Table before the data are included on the workload measures.

Calculate Subject Density

The Calculate Subject Density element, shown in Figure 4-6, first processes density at the cell level and then at the block level. A local table, SUBJECT_DENSITY, is used to store the cell occupancy of the subject aircraft. The aircraft count field of this table is initialized to zero. For each cell (in x, y, t) in the grid chain (SPARSE_CELLS), the corresponding records in the sector airspace grid (SWP_CELL) are examined where the x and y dimensions have the same values. Providing

```
ROUTINE Calculate_Subject_Density;
REFER TO GLOBAL SWP_CELL IN, SPARSE_CELLS IN;
REFER TO SHARED LOCAL Sflid IN, SUBJECT_CELL INOUT, SUBJECT_BLOCK
   INOUT, CELL_DENSITY INOUT, BLOCK_DENSITY INOUT;
DEFINE VARIABLES
   Start_Cell_Time   Beginning time for x,y,t cell
   Cell_Count        Number of cells occupied by the subject
                        aircraft in one block;
```

FIGURE 4-6
CALCULATE_SUBJECT_DENSITY

```
#processing at the cell level#
   #store cell occupancy of subject aircraft in SUBJECT_CELL#
REPEAT FOR EACH SPARSE_CELLS RECORD #in x,y,t#
#For given values of x,y in SPARSE_CELLS repeat for z in SWP_CELL#
   WHERE SPARSE_CELLS.fl_id EQ Sflid;
   REPEAT FOR EACH SWP_CELL RECORD #in x,y,z#
      WHERE x,y dimensions of SWP_CELL correspond to x,y dimensions
         of SPARSE_CELLS; #range of z in center is covered #
      IF interval (SPARSE_CELLS.min_z, SPARSE_CELLS.max_z) and
         interval (SWP_CELL.min_altitude, SWP_CELL.max_altitude)
         overlap
      THEN #aircraft occupies this x,y,z cell#
         Calculate Start_Cell_Time from time part of SPARSE_CELLS.
            tree_node_id;
         #add cell occupancy of subject aircraft (to SUBJECT_#
            #CELL and to CELL_DENSITY)#
         UPDATE IN CELL_DENSITY (aircraft_count =aircraft_count + 1)
            WHERE CELL_DENSITY.cell_id EQ SWP_CELL.cell_id AND
               CELL_DENSITY.beg_time EQ Start_Cell_Time;
         UPDATE IN SUBJECT_CELL (aircraft_count = aircraft_count
            + 1)
            WHERE SUBJECT_CELL.cell_id EQ SWP_CELL.cell_id AND
               SUBJECT_CELL.beg_time EQ Start_Cell_Time AND
               SUBJECT_DENSITY.fl_id EQ Sflid;
#Processing at the block level#
#Use cell occupancy of subject to get block occupancy#
REPEAT FOR EACH BLOCK_DENSITY RECORD;
   SELECT FIELDS ALL
      RETURN COUNT (Cell_Count)
      FROM SUBJECT_CELL
      WHERE SUBJECT_CELL.aircraft_count GT 0 AND SUBJECT_CELL.
         block_id EQ BLOCK_DENSITY.block_id AND SUBJECT_ CELL.beg_
         time is within the time interval defined by BLOCK
         DENSITY.beg_time AND SUBJECT_DENSITY.fl_id EQ Sflid;
   IF Cell_Count GT 0
   THEN
      UPDATE IN BLOCK_DENSITY (aircraft_count = aircraft_count + 1);
      UPDATE IN SUBJECT_BLOCK (aircraft_count = aircraft_count + 1)
         WHERE SUBJECT_BLOCK.fl_id EQ Sflid AND SUBJECT_BLOCK.
            block_id EQ BLOCK_DENSITY.block_id AND SUBJECT_
            BLOCK.beg_time EQ BLOCK_DENSITY.beg_time;
END Calculate_Subject_Density;
```

FIGURE 4-6
CALCULATE_SUBJECT_DENSITY (Concluded)

```
ROUTINE Calculate_Sector;
#routine called during processing of one SPARSE_CELLS record#
#this routine determines the basic sector which is associated with#
#one cell of SPARSE_CELLS (in x,y,t) based on the sector with the#
#greatest change in altitude over the x,y,t cell#
PARAMETERS CELL IN, Basic_Sector_Number OUT;
REFER TO GLOBAL SWP_CELL IN, SPARSE_CELLS IN;
DEFINE VARIABLES
   Basic_Sector_Number  Basic sector for x,y cell
   Alt_Overlap          Overlap of altitude for subject with x,y,z
   No_Sectors           Number of sectors associated with CELL;
DEFINE TABLES
   CELL                 Contains information on one cell in the grid
                           chain (SPARSE_CELLS); fields defined like
                           SPARSE_CELLS
   POSS_SECTORS         Possible sectors associated with CELL.cell_id
      cell_id              Cell in x,y,z
      diff_alt             Altitude overlap
      sector_number        Basic sector;
   REPEAT FOR EACH SWP_CELL RECORD #in x,y,z#
      WHERE x,y dimensions of SWP_CELL correspond to x,y dimensions of
         CELL; #repeat for z divisions in SWP_CELL for x,y #
      IF interval (CELL.min_z, CELL.max_z) and interval (SWP_CELL.
         min_altitude, SWP_CELL.max_altitude) overlap
      THEN
         #intersection of altitude ranges#
         Alt_Overlap = amount of overlap;
         INSERT INTO POSS_SECTORS (cell_id = SWP_CELL.cell_id, diff_
            alt = Alt_Overlap, sector_number = SWP_CELL.sector_
            number);
   No_Sectors = COUNT (unique values of POSS_SECTORS.sector_number);
   IF No_Sectors EQ 1
   THEN
      Basic_Sector_Number = unique POSS_SECTORS.sector_number;
   ELSE
      Basic_Sector_Number = sector with the maximum sum of POSS_
         SECTORS.diff_alt;
END Calculate_Sector;
```

FIGURE 4-7
CALCULATE_SECTOR

```
ROUTINE Determine_Unique_Sector_Time_Intervals;
REFER TO SHARED LOCAL REVISED_SUBJECT_WORKLOAD INOUT;
DEFINE VARIABLES
   Min_Time                      Minimum of beginning times examined
   Max_Time                      Maximum of ending times examined
   End_Time (*)                  Array of unique ending times in RSW
   Beg_Time (*)                  Array of unique beginning times in
                                 RSW;
DEFINE TABLES
   SEC_TI              Sector time interval table
      sector_number       Basic sector
      time_interval_id    Corresponding time interval;
   SEC_TI = SELECT FIELDS UNIQUE sector_number, time_interval_id
              FROM REVISED_SUBJECT_WORKLOAD; #unique field pairs#
   REPEAT FOR EACH SEC_TI RECORD; #determine beginning time and#
      #ending time for each sector time interval#
      #determine beginning time of sector time interval#
      SELECT FIELDS UNIQUE beginning_time
         FROM REVISED_SUBJECT_WORKLOAD (RSW)
         INTO Beg_Time  #array#
         WHERE RSW.sector_number EQ SEC_TI.sector_number AND
            RSW.time_interval_id EQ SEC_TI.time_interval_id;
      Min_Time = MIN(Beg_Time); #MIN is a builtin function#
      #determine ending time of sector time interval#
      SELECT FIELDS UNIQUE ending_time
         FROM REVISED_SUBJECT_WORKLOAD (RSW)
         INTO End_Time #array#
         WHERE RSW.sector_number EQ SEC_TI.sector_number AND RSW.
            time_interval_id EQ SEC_TI.time_interval_id;
      Max_Time = MAX (End_Time); #MAX is a builtin function#
      DELETE FROM REVISED_SUBJECT_WORKLOAD (RSW) #delete records in#
         #RSW for x, y, z, t cells related to sector time interval#
         WHERE RSW.sector_number EQ SEC_TI.sector_number AND RSW.
            .time_interval_id EQ SEC_TI.time_interval_id;
      INSERT INTO REVISED_SUBJECT_WORKLOAD
         #add 1 record in RSW for sector time interval#
         (sector_number = SEC_TI.sector_number, time_interval_id =
         SEC_TI.time_interval_id, beginning_time = Min_Time, ending_
         time = Max_Time, total_fl_time = 0, FPCP_encounter_count =
         0, AP_encounter_count = 0, pa_counts = 0);
   #end of sector time interval logic#
END Determine_Unique_Sector_Time_Intervals;
```

FIGURE 4-8
DETERMINE_UNIQUE_SECTOR_TIME_INTERVALS

the altitude for the grid chain cell overlaps that for the sector airspace grid cell, the aircraft's occupancy is included in the cell density airspace grid (CELL_DENSITY) and in SUBJECT_DENSITY. Next, processing of density at the block level uses the cell occupancy of the subject to add to the block occupancy of all aircraft. For each block density airspace grid (BLOCK_DENSITY) block, a count (Cell_Count) is made for the number of occupied cells in the cell density airspace grid corresponding to this block. If Cell_Count is greater than zero, then the aircraft count field in BLOCK_DENSITY is incremented.

Calculate Sector

The sector associated with a cell in the subject's grid chain is determined by the Calculate Sector element displayed in Figure 4-7. The cells in the sector airspace grid (SWP_CELL) which have values in the x and y dimensions equal to those in the subject's grid chain cell are examined to see if the altitude intervals of these cells overlap. If the altitude intervals overlap, the cell id, sector number and amount of overlap are stored in POSS_SECTORS. For one cell in the grid chain, the aircraft may be in more than one cell in the sector airspace grid; therefore, the number of unique sectors in POSS_SECTORS is counted. If the number of unique sectors is not equal to one, the sector corresponding to the grid chain cell is determined to be the sector with the maximum sum of altitude overlap in POSS_SECTORS.

Determine Unique Sector Time Intervals

The Determine Unique Sector Time Intervals element, described in Figure 4-8, processes and consolidates information in the RSW Table such that this table is output containing only one record for each sector-time-interval. First the unique pairs of sector number and time-interval id are selected into a local table (SEC_TI). For each record in SEC_TI, the beginning time and ending time are computed from records in RSW with the same sector and time-interval. The beginning time is calculated as the minimum of the Beg_Time in these RSW records; the ending time is computed to be the maximum of the End_Time in these records. Next, the records in the RSW Table with the same sector number and time-interval are deleted and one new record is added with the beginning time and ending time for the entire time-interval.

Include Subject Encounters

After the Create Subject Sector Time Interval element is completed, the Include Subject Encounters element described in Figure 4-9 is processed. The Include Subject Encounters element processes separately the FPCP encounters and AP encounters for the subject and updates the encounter counts in the appropriate sector-time-interval in the RSW Table. For the FPCP encounters, the display-as-advisory time in ENCOUNTERS is compared to the sector-time-interval beginning time and ending time. For the AP encounters, the display-as-advisory time is computed as the current time or the time of penetration of the static airspace subtract the AP_Not_Dur. Similarly, the display-as-advisory time for the AP encounters is compared to the sector-time-interval beginning time and ending time.

Include Object Encounters

The FPCP encounters for the subject affect some object aircraft--specifically the ESC aircraft. The purpose of the Include Object Encounters element, displayed in Figure 4-10, is to designate the ESC aircraft and to update the FPCP encounter count for their records in the IAW Table. First the records in ENCOUNTERS for the current encounters are processed. The FPCP encounter count in the new sector-time-interval is incremented by one. Information on this ESC aircraft is inserted in the ESC_INCREMENT Table. Secondly, the records in PRIOR_ENCOUNTERS for the previous encounters are processed. The FPCP encounter count in the former sector-time-interval is decremented by one and data on this object are inserted in the ESC_DECREMENT Table. Two ESC Tables have been created which will be used by the Basic Sector Workload Update component to update the BASIC_SECTOR_WORKLOAD_MEASURES Table.

Include Subject Planned Actions

Data on another workload measure--the planned actions--are incorporated in the RSW Table by Include Subject Planned Actions element depicted in Figure 4-11. Each planned action in PLANNED_ACTIONS for the subject is processed. The start time of the planned action is obtained from PLANNED_ACTION_DURATION for the corresponding planned action. Next the sector-time-interval in RSW which contains the planned action start time is located and the count for this planned action type is incremented by one.

```
ROUTINE Include_Subject_Encounters;
REFER TO GLOBAL ENCOUNTERS IN, ENVIRONMENTAL_CONFLICT IN;
REFER TO SHARED LOCAL REVISED_SUBJECT_WORKLOAD INOUT, AP_Not_Dur IN,
   Sflid IN;
   #Process FPCP Encounters#
   REPEAT FOR EACH ENCOUNTERS RECORD
      WHERE ENCOUNTERS.first_fl_id EQ Sflid OR ENCOUNTERS.second_fl_
         id EQ Sflid;
      REPEAT FOR EACH REVISED_SUBJECT_WORKLOAD (RSW) RECORD
         WHERE RSW.beginning_time LE ENCOUNTERS.display_as_advisory_
            time AND RSW.ending_time GT ENCOUNTERS.display_as_
            advisory_time;
         UPDATE IN REVISED_SUBJECT_WORKLOAD (FPCP_encounter_count =
            FPCP_encounter_count + 1);
      #Process AP Encounters#
   REPEAT FOR EACH ENVIRONMENTAL_CONFLICT RECORD
      WHERE ENVIRONMENTAL_CONFLICT.fl_id EQ Sflid;
      REPEAT FOR EACH REVISED_SUBJECT_WORKLOAD (RSW) RECORD
         WHERE RSW.beginning_time LE (ENVIRONMENTAL_CONFLICT.time -
            AP_Not_Dur) AND RSW.ending_time GT (ENVIRONMENTAL_
            CONFLICT.time - AP_Not_Dur);
         UPDATE IN REVISED_SUBJECT_WORKLOAD (AP_encounter_count =
            AP_encounter_count + 1);
END Include_Subject_Encounters;
```

FIGURE 4-9
INCLUDE_SUBJECT_ENCOUNTERS

```
ROUTINE Include_Object_Encounters;
REFER TO GLOBAL ENCOUNTERS IN, PRIOR_ENCOUNTERS IN;
REFER TO SHARED LOCAL ESC_INCREMENT OUT, ESC_DECREMENT OUT,
    INDIVIDUAL_AIRCRAFT_WORKLOAD INOUT, Sflid IN, Trajectory_Update_
    Status IN;
    REPEAT FOR EACH ENCOUNTERS RECORD
        #add FPCP encounters for object to ESC_INCREMENT table#
        WHERE ENCOUNTERS.first_fl_id EQ Sflid OR ENCOUNTERS.second_fl_id
            EQ Sflid;
        #Note hereafter first_fl_id represents the subject's flid, #
            #second_fl_id represents object's flid#
        REPEAT FOR EACH INDIVIDUAL_AIRCRAFT_WORKLOAD (IAW) RECORD
            WHERE IAW.fl_id EQ ENCOUNTERS.second_fl_id AND IAW.beginning_
                time LE ENCOUNTERS.display_as_advisory_time AND IAW.
                ending_time GT ENCOUNTERS.display_as_advisory_time;
            UPDATE IN INDIVIDUAL_AIRCRAFT_WORKLOAD (FPCP_encounter_count
                = FPCP_encounter_count + 1);
            INSERT INTO ESC_INCREMENT (fl_id = INDIVIDUAL_AIRCRAFT_
                WORKLOAD.fl_id, sector_number = INDIVIDUAL_AIRCRAFT_
                WORKLOAD.sector_number, time_interval_id = INDIVIDUAL_
                AIRCRAFT_WORKLOAD.time_interval_id);
    IF Trajectory_Update_Status EQ 'REVISED'#not a new trajectory#
    THEN #remove previous FPCP encounters for subject from IAW table#
        #and add previous FPCP encounters for objects to ESC_DECREMENT#
            #table#
        REPEAT FOR EACH PRIOR_ENCOUNTERS RECORD
            WHERE PRIOR_ENCOUNTERS.first_fl_id EQ Sflid OR PRIOR_
                ENCOUNTERS.second_fl_id EQ Sflid;
            REPEAT FOR EACH INDIVIDUAL_AIRCRAFT_WORKLOAD (IAW) RECORD
                WHERE IAW.fl_id EQ PRIOR_ENCOUNTERS.second_fl_id AND IAW.
                    beginning_time LE PRIOR_ENCOUNTERS.display_as_advisory_
                    time AND IAW.ending_time GT PRIOR_ ENCOUNTERS.display_
                    as_advisory_time;
                UPDATE IN INDIVIDUAL_AIRCRAFT_WORKLOAD (FPCP_encounter_
                    count = FPCP_encounter_count - 1);
                INSERT INTO ESC_DECREMENT (fl_id = INDIVIDUAL_AIRCRAFT_
                    WORKLOAD.fl_id, sector_number = INDIVIDUAL_AIRCRAFT_
                    WORKLOAD.sector_number, time_interval_id = INDIVIDUAL_
                    AIRCRAFT_WORKLOAD.time_interval_id);
        DELETE FROM PRIOR_ENCOUNTERS;  #entire table#
END Include_Object's_Encounters;
```

FIGURE 4-10
INCLUDE_OBJECT_ENCOUNTERS

```
ROUTINE Include_Subject_Planned_Actions;
REFER TO GLOBAL PLANNED_ACTIONS IN, PLANNED_ACTION_DURATION IN;
REFER TO SHARED LOCAL REVISED_SUBJECT_WORKLOAD INOUT, Sflid IN;
DEFINE VARIABLES
   Pa_Start_Time  Time planned action being processed begins;
   #type of pa count refers to the count on one of the pa types in#
      #RSW, i.e., whichever one is appropriate#
   REPEAT FOR EACH PLANNED_ACTIONS RECORD
      WHERE PLANNED_ACTIONS.fl_id EQ Sflid;
      SELECT FIELDS pa_start_time
         FROM PLANNED_ACTION_DURATION
         INTO Pa_Start_Time
         WHERE PLANNED_ACTION_DURATION.pa_id EQ PLANNED_ACTIONS.pa_id;
      REPEAT FOR EACH REVISED_SUBJECT_WORKLOAD (RSW) RECORD
         WHERE RSW.beginning_time LE Pa_Start_Time AND RSW.ending_time
            GT Pa_Start_Time;
         UPDATE IN REVISED_SUBJECT_WORKLOAD (increment type of pa
            count by 1 for which the type of pa count is the PLANNED_
            ACTIONS.pa_type);
END Include_Subject_Planned_Actions;
```

FIGURE 4-11
INCLUDE_SUBJECT_PLANNED_ACTIONS

4.1.2 Basic Sector Workload Update Component

4.1.2.1 Mission

The purpose of the Basic Sector Workload Update component is to update the workload measures in the BASIC_SECTOR_WORKLOAD_MEASURES Table, to reflect the changes due to a trajectory update. The IAW Table records for the subject aircraft are also updated or created.

4.1.2.2 Design Considerations and Component Environment

Input

The input data contain the IAW Table, the RSW Table, ESC Tables, BASIC_SECTOR_WORKLOAD_MEASURES Table, the subject flight identifier, and the status of the trajectory update (new or revised). The BASIC_SECTOR_WORKLOAD_MEASURES Table has the data on the aggregate aircraft according to sectors and time-intervals, for all basic sectors in the center and for all time-intervals within the time horizon.

Output

The primary output is the BASIC_SECTOR_WORKLOAD_MEASURES Table updated to reflect the trajectory update, including, for each time-interval and sector pair:

- Total flight time
- Number of planned actions according to type
- Number of encounters from AP and FPCP

The subject's records in the IAW Table are also updated.

4.1.2.3 Component Design Logic

The Basic Sector Workload Update logic, shown in Figure 4-12, begins with the addition of the data from the RSW Table to the BASIC_SECTOR_WORKLOAD_MEASURES (BSWM) Table. For each sector-time-interval in the RSW Table, the corresponding sector-time-intervals in the BSWM Table are examined to determine which sector and time-interval are affected. For each such pair, the following RSW Table values are added to the corresponding BSWM Table values: aircraft minutes, counts on planned actions, and count on Airspace Probe and Flight Plan Conflict Probe encounters.

```
ROUTINE Allobject_Workload_Update;
REFER TO GLOBAL BASIC_SECTOR_WORKLOAD_MEASURES INOUT;
REFER TO SHARED LOCAL REVISED_SUBJECT_WORKLOAD IN, INDIVIDUAL_
   AIRCRAFT_WORKLOAD INOUT, Sflid IN, Trajectory_Update_Status IN,
   ESC_INCREMENT IN, ESC_DECREMENT IN;
#                                                                    #
# In this routine, for purposes of clarity and simplicity, the #
   # following abbreviations are used: #
   # RSW for REVISED_SUBJECT_WORKLOAD #
   # IAW for INDIVIDUAL_AIRCRAFT_WORKLOAD #
```

FIGURE 4-12
BASIC_SECTOR_WORKLOAD_UPDATE

```
REPEAT FOR EACH REVISED_SUBJECT_WORKLOAD (RSW) RECORD;
   #add workload for subject to that for all aircraft#
   UPDATE IN BASIC_SECTOR_WORKLOAD_MEASURES (BSWM)(total_fl_time =
      total_fl_time + RSW.total_fl_time, fp_conflict_count = fp_
      conflict_count + RSW.FPCP_encounter_count, airspace_conflict_
      count = AP_conflict_count + RSW.AP_encounter_count, pa_counts =
      RSW.pa_counts);
      WHERE BSWM.time_interval_id EQ RSW.time_interval_id AND
         BSWM.sector_number EQ RSW.sector_number;
IF Trajectory_Update_Status EQ 'REVISED'.
THEN
   REPEAT FOR EACH INDIVIDUAL_AIRCRAFT_WORKLOAD (IAW) RECORD
   #subtract previous workload for subject from that for all aircraft#
      WHERE IAW.fl_id EQ Sflid;
      UPDATE IN BASIC_SECTOR_WORKLOAD_MEASURES (BSWM)(total_fl_time =
         total_fl_time - IAW.total_fl_time, fp_conflict_count = fp_
         conflict_count - IAW.FPCP_encounter_count, airspace_
         conflict_count = airspace_conflict_count - IAW.AP_encounter_
         count, pa_counts = pa_counts - IAW.pa_counts)
         WHERE BSWM.sector_number EQ IAW.sector_number AND BSWM.time_
            interval_id EQ IAW.time_interval_id;
   REPEAT FOR EACH ESC_INCREMENT RECORD  #add to FPCP encounters for#
      #objects#
      UPDATE IN BASIC_SECTOR_WORKLOAD_MEASURES (BSWM) (fp_conflict_
         count = fp_conflict_count + 1)
         WHERE BSWM.sector_number EQ ESC_INCREMENT.sector_number AND
            BSWM.time_interval_id EQ ESC_INCREMENT.time_interval_id;
   REPEAT FOR EACH ESC_DECREMENT RECORD  #decrease number of FPCP #
      #encounters for objects#
      UPDATE IN BASIC_SECTOR_WORKLOAD_MEASURES (BSWM) (fp_conflict_
         count = fp_conflict_count - 1)
         WHERE BSWM.sector_number EQ ESC_DECREMENT.sector_number AND
            BSWM.time_interval_id EQ ESC_DECREMENT.time_interval_id;
   DELETE FROM INDIVIDUAL_AIRCRAFT_WORKLOAD (IAW)
      #remove subject's previous workload#
      WHERE IAW.fl_id EQ Sflid;
REPEAT FOR EACH REVISED_SUBJECT_WORKLOAD RECORD
   #save subject's new workload#
   INSERT INTO INDIVIDUAL_AIRCRAFT_WORKLOAD (IAW) (fl_id = Sflid,
      sector_number = RSW.sector_number, time_interval_id = RSW.time_
      interval_id, beginning_time = RSW.beginning_time, ending_time =
      RSW.ending_time, total_fl_time = RSW.total_fl_time, FPCP_
      encounter_count = RSW.FPCP_encounter_count, AP_encounter_count
      = RSW.AP_encounter_count, pa_counts = RSW.pa_counts);
END Allobject_Workload_Update;
```

FIGURE 4-12

BASIC_SECTOR_WORKLOAD_UPDATE (Concluded)

If the status of the trajectory update indicates that a revision to the trajectory has occurred, then a process similar to that described above is performed with the subtraction of the values in the subject's records of the IAW Table from the corresponding BASIC_SECTOR_WORKLOAD_MEASURES Table values.

The updating of the BASIC_SECTOR_WORKLOAD_MEASURES Table counts for FPCP encounters is not yet complete: The old workload information has been deleted and the new workload information added for the subject but not yet for the ESC objects. To do so, the algorithm uses the ESC Tables to subtract the old encounter counts and add the new ones to the BASIC_SECTOR_WORKLOAD_MEASURES Table.

The data in the subject's records in the IAW Table (reflecting a previous trajectory update) are no longer needed and are therefore deleted. The RSW Table data are inserted in the IAW Table.

4.2 Supervisor Requests Subfunction

The Supervisor Requests subfunction is invoked by the supervisor for one of the following reasons:

- To display the outputs for the sectors in this supervisor's area or any set of sectors in the planning region (Immediate Mode Request or Conditional Mode Request)

- To set a threshold on a sector and workload measure (Conditional Mode Request)

- To edit the display features of the outputs

In this subfunction, workload is handled in terms of aggregate aircraft categorized by sectors and time-intervals, using the BASIC_SECTOR_WORKLOAD_MEASURES Table but not the INDIVIDUAL_AIRCRAFT_WORKLOAD Table.

The values of several user-selected variables are required in order that the Supervisor Requests subfunction can set the values of the corresponding shared local parameters described in Appendix A. These user-selected variables are as follows:

- Request_Type
- Organization Type (Organ_Type)
- Periodic_Frequency
- Area_Name
- List_Of_Sectors

- Ac_Count_Option
- Weighted_Pa_Option
- Density_Option
- Overall_Workload_Option

Default values may be used for some of these variables, if not input by the supervisor.

The Supervisor Requests subfunction also requires as input the sectorization plan (SECTORIZATION_PLAN) and the sectorization schedule (SECTORIZATION_SCHEDULE). The values in the SECTORIZATION_PLAN Table are updated infrequently, but the values in the SECTORIZATION_SCHEDULE Table are updated by the supervisor as the schedule changes.

The outputs of the Supervisor Requests subfunction include the BASIC_SECTOR_WORKLOAD_MEASURES Table, the COMBINED_SECTOR_WORKLOAD_MEASURES Table, the WORKLOAD_THRESHOLDS Tables, and the values of the display options.

The components of Supervisor Requests subfunction are as follows:

- Workload Evaluation
- Threshold Request
- Display Features

No calling sequence is specified for this subfunction. These components operate independently of each other. Figure 4-13 illustrates the association of the components with the Supervisor Requests subfunction.

4.2.1 Workload Evaluation Component

4.2.1.1 Mission

The Workload Evaluation component performs the final calculations of the workload measures, applies the weighting coefficients, and computes the overall workload values for the sectors in the area or the sector(s) requested. The component is activated either directly by the supervisor, with an Immediate Mode Request, or automatically by SWP, at periodic intervals, following a Conditional Mode Request.

4.2.1.2 Design Considerations and Component Environment

The sectorization schedule that is in effect over the display time horizon is considered by this component. If a new

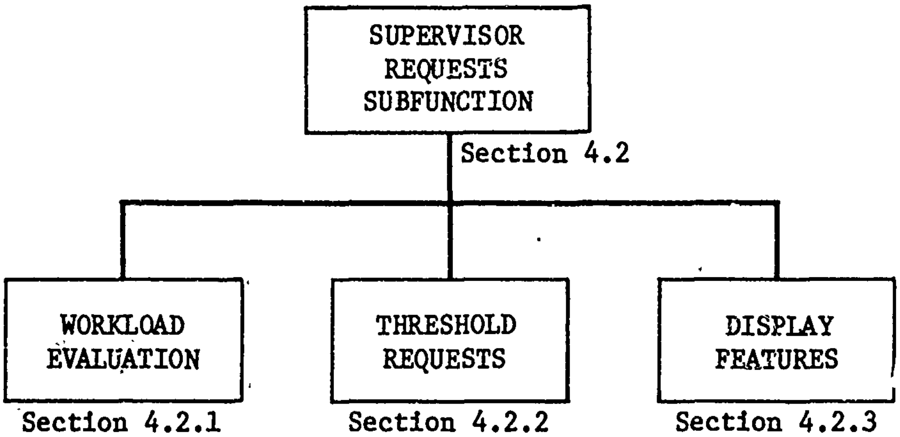


FIGURE 4-13
SUPERVISOR REQUESTS SUBFUNCTION

sectorization plan becomes effective during a time-interval, then the workload values are calculated for the sectorization plan which is active the larger percent of the time for that time-interval.

Input

Input data for the Workload Evaluation component contain both global and shared local tables and parameters. The global data for this component consist of the following which are described in "Data Specification for AERA 1.0" [Vol. 5]:

- BASIC_SECTOR_WORKLOAD_MEASURES
- Sectorization Plan (SECTORIZATION_PLAN)
- Sectorization Schedule (SECTORIZATION_SCHEDULE)
- Display_Time_Horizon
- Time_Interval_Duration
- Cell_Density_Ratio
- Aircraft Count Coefficient (Ac_Coefficient)
- Flight Plan Conflict Coefficient (Flight_Plan_Cfl_Coefficient)
- Airspace Coefficient (Airspace_Cfl_Coefficient)
- Pa_Coefficients
- Density_Coefficient

The shared local data for this component include the following tables and parameters described in Appendix A.

- Cell Density Airspace Grid (CELL_DENSITY)
- Block Density Airspace Grid (BLOCK_DENSITY)
- Organization Type (Organ_Type)
- List_Of_Sectors
- Area_Name
- Aircraft Count Option (Ac_Count_Option)
- Weighted_Pa_Option
- Density_Option
- Overall_Workload_Option
- Display_Time_Intervals

Output

The outputs are the five workload measures according to sector and time-interval, as shown in Section 3.1.2.2. Depending on the value of the organization type variable, these measures are output in either the BASIC_SECTOR_WORKLOAD_MEASURES Table or the COMBINED_SECTOR_WORKLOAD_MEASURES Table.

4.2.1.3 Component Design Logic

Workload Evaluation

The Workload Evaluation component processes data only for the specified measure(s), time interval(s) and sector(s).

As Figure 4-14 illustrates, the Workload Evaluation component tests whether the output is associated with an organization type (Organ_Type) of sector or area. If the Organ_Type is 'SECTOR,' then the Calculate Basic Sector Workload element is invoked for each sector in List_of_Sectors. However, if the Organ_Type is 'AREA,' then the Calculate Area Workload element is invoked. The Workload Evaluation component contains the following elements which are called in sequence depending upon whether the Calculate Basic Sector Workload element or Calculate Area Workload element is invoked:

- Calculate Basic Sector Workload
  - Compute Basic Sector Density
    - Compute Unit Density Sum
      - Compute Percent of Aircraft
- Calculate Area Workload
  - Compute Basic Sector Density
  - Compute Combined Sector Calculations

The workload measures may be adjusted (if the supervisor specifies) by the adaptation factors.

Calculate Basic Sector Workload

The Calculate Basic Sector Workload element, shown in Figure 4-15, computes the workload on one sector for the display time-intervals requested. For each appropriate record in the BASIC_SECTOR_WORKLOAD_MEASURES (BSWM) Table, the options on various workload measures are tested to determine if their workload values need to be computed. If the value of the overall workload measure is computed, then the values of other measures are automatically computed. Finally the values of these workload measures are updated in the BSWM Table.

The main portion of the calculations for the density measure is contained in three elements—Compute Basic Sector Density, Compute Unit Density Sum and Compute Percent of Aircraft. These elements are described in Figure 4-16, 4-17, and 4-18.

The cell density airspace grid and the block density airspace grid contain the aircraft count for density on the cell level

```
ROUTINE Workload_Evaluation;
REFER TO GLOBAL BASIC_SECTOR_WORKLOAD_MEASURES OUT, COMBINED_SECTOR_
   WORKLOAD_MEASURES OUT;
REFER TO SHARED LOCAL List_Of_Sectors IN;
DEFINE VARIABLES
   I                    Index
   Basic_Sector_Number  Basic sector id for sector to be displayed;
   IF Organ_Type EQ 'SECTOR'
   THEN
      FOR I = 1 TO COUNT (List_Of_Sectors);
         Basic_Sector_Number = List_Of_Sectors(I);
         CALL Calculate_Basic_Sector_Workload (Basic_Sector_Number
            IN);
   ELSE #it is an 'AREA'#
      CALL Calculate_Area_Workload;
END Workload Evaluation;
```

FIGURE 4-14
WORKLOAD EVALUATION

```
ROUTINE Calculate_Basic_Sector_Workload;
#calculate for one basic sector#
PARAMETERS Basic_Sector_Number IN;
REFER TO GLOBAL BASIC_SECTOR_WORKLOAD_MEASURES INOUT, Display_Time_
   Horizon IN, Time_Interval_Duration IN; Pa_Coefficients IN, Cell_
   Density_Ratio IN, Ac_Coefficient IN, Flight_Plan_Cfl_Coefficient
   IN, Airspace_Coefficient IN, Density_Coefficient IN;
REFER TO SHARED LOCAL Ac_Count_Option IN, Weighted_Pa_Option IN,
   Density_Option IN, Overall_Workload_Option IN, Display_Time_
   Intervals IN;
DEFINE VARIABLES
   Basic_Sector_Number  Basic sector number for sector to be
                           displayed
   Cell_Density_Value   Sum of percent of aircraft for cell
                           density for sector time interval
   Block_Density_Value  Sum of percent aircraft for block
                           density for sector time interval
   Den                  Density value for sector time interval
                           being processed
   Ac_Count             A value of the average number of
                           aircraft for the sector time interval
                           being processed
   Wpa                  Weighted combined pa value for sector time
                           interval being processed
   Overall              Overall workload value for sector time
                           interval being processed;
```

FIGURE 4-15
CALCULATE_BASIC_SECTOR_WORKLOAD

```
REPEAT FOR EACH BASIC_SECTOR_WORKLOAD_MEASURES (BSWM) RECORD
   WHERE (BSWM.sector_number EQ Basic_Sector_Number) AND (BSWM.time_
      interval_id indicates that the time is in the Display_Time_
      Intervals and is also between the current time and the Display_
      Time_Horizon);
   Wpa = 0;
   Den = 1;
   Overall = 0;
   Ac_Count = 0;
   IF (Density_Option EQ TRUE) OR (Overall_Workload_Option EQ TRUE)
   THEN #compute density#
      CALL Compute_Basic_Sector_Density (Cell_Density_Value OUT,
         Block_Density_Value OUT, Basic_Sector_Number IN);
      Den = Cell_Density_Ratio * ((Cell_Density_Value/3 - 0.283)/(1 -
         0.283)) + (1 - Cell_Density_Ratio) * ((Block_Density_Value/3
         - 0.283)/(1 - 0.283));
   IF (Ac_Count_Option EQ TRUE) OR (Overall_Workload_Option EQ TRUE)
   THEN #compute average aircraft count#
      Ac_Count = BASIC_SECTOR_WORKLOAD_MEASURES.total_fl_time/Time_
      Interval_Duration;
   IF (Weighted_Pa_Option EQ TRUE) OR (Overall_Workload_Option EQ
      TRUE)
   THEN #compute weighted planned action value#
      Wpa = SUM (BASIC_SECTOR_WORKLOAD_MEASURES.pa_counts * Pa_
         Coefficients);
   IF Overall_Workload_Option EQ TRUE
   THEN #compute overall workload#
      Overall = Ac_Count * Ac_Coefficient + BASIC_SECTOR_WORKLOAD_
         MEASURES.fp_conflict_count * Flight_Plan_Cfl_Coefficient +
         BASIC_SECTOR_WORKLOAD_MEASURES.airspace_conflict_count *
         Airspace_Cfl_Coefficient + Wpa + Den * Density_Coefficient;
   UPDATE IN BASIC_SECTOR_WORKLOAD_MEASURES (aver_aircraft_count =
      Ac_Count, weighted_pa = Wpa, density = Den, overall_workload_
      measures = Overall);
END Calculate_Basic_Sector_Workload;
```

FIGURE 4-15
CALCULATE_BASIC_SECTOR_WORKLOAD (Concluded)

```
ROUTINE Compute_Basic_Sector_Density;
PARAMETERS Cell_Density_Value OUT, Block_Density_Value OUT, Basic_
   Sector_Number IN;
REFER TO GLOBAL Time_Interval_Duration IN;
REFER TO SHARED LOCAL CELL_DENSITY IN, BLOCK_DENSITY IN, DELTA_T IN;
DEFINE VARIABLES
   Cell_Density_Value    Sum of percent of aircraft for cell
                            density for sector time interval
   Block_Density_Value   Sum of percent of aircraft for block
                            density for sector time interval
   Basic_Sector_Number   Basic sector being processed
   Tot_Ac_Count          Total number of aircraft in the sector
                            time interval
   Start_Cell_Time       Beginning time associated with time
                            division
   J                     Index;
DEFINE TABLES
   UNIT_D                Table for unit of cells or blocks
      unit_id               Cell or block identifier
      aircraft_count        Number of aircraft in the unit;
```

FIGURE 4-16
COMPUTE_BASIC SECTOR_DENSITY

```
#Compute density on cell level; UNIT_D used for cells#
Tot_Ac_Count = 0;
#Step through time divisions in intervals#
FOR J = 1 TO Time_Interval_Duration/Delta_T;
   Start_Cell_Time = the time at the beginning of time division J;
   #Step through cells in sector and add the number of aircraft in#
   #each cell to the total#
   REPEAT FOR EACH CELL_DENSITY RECORD
      WHERE (CELL_DENSITY.sector_number EQ Basic_Sector_Number) AND
         (CELL_DENSITY.beg_time EQ Start_Cell_Time);
      IF J EQ 1
      THEN  #this is the first time division#
         INSERT INTO UNIT_D (unit_id = CELL_DENSITY.cell_id,
            aircraft_count = CELL_DENSITY.aircraft_count);
      #Otherwise, add the number of aircraft in the cell to the saved#
      #number for that cell#
      ELSE
         UPDATE IN UNIT_D (aircraft_count = aircraft_count + CELL_
            DENSITY.aircraft_count)
         WHERE UNIT_D.unit_id EQ CELL_DENSITY.cell_id;
      Tot_Ac_Count = Tot_Ac_Count + CELL_DENSITY.aircraft_count;
CALL Compute_Unit_Density_Sum (Cell_Density_Value OUT, Tot_Ac_Count
   IN, UNIT_D IN);
DELETE FROM UNIT_D; #delete entire table to use again#
#Compute density on block level; UNIT_D used for block density#
Tot_Ac_Count = 0;
FOR J = 1 TO Time_Interval_Duration/Delta_T;
   Compute Start_Cell_Time for time division J;
   REPEAT FOR EACH BLOCK_DENSITY RECORD #each block in sector#
      WHERE (BLOCK_DENSITY.sector_number EQ Basic_Sector_Number) AND
         (BLOCK_DENSITY.beg_time EQ Start_Cell_Time);
      IF J EQ 1
      THEN
         INSERT INTO UNIT_D (unit_id = BLOCK_DENSITY.block_id,
            aircraft_count = BLOCK_DENSITY.aircraft_count);
      ELSE
         UPDATE IN UNIT_D (aircraft_count = aircraft_count + BLOCK_
            DENSITY.aircraft_count)
            WHERE UNIT_D.unit_id EQ BLOCK_DENSITY.block_id;
      Tot_Ac_Count = Tot_Ac_Count + BLOCK_DENSITY.aircraft_count;
CALL Compute_Unit_Density_Sum (Block_Density_Value OUT, Tot_Ac_Count
   IN, UNIT_D IN);
END Compute_Basic_Sector_Density;
```

FIGURE 4-16

COMPUTE_BASIC_SECTOR_DENSITY (Concluded)

```
ROUTINE Compute_Unit_Density_Sum;
PARAMETERS Sum_Per_Ac OUT, Tot_Ac_Count IN, UNIT_D IN;
DEFINE VARIABLES
   Per_Cell_Inv       Inverse of Per_Cell (proportions of sector's
                         cells with the greatest number of aircraft)
   Per_Ac             Percent of aircraft corresponding to Per_Cell
   Sum_Per_Ac         Accumulated sum of Per_Ac
   Tot_Ac_Count       Number of aircraft in the sector
   Tot_No_Cells       Total number of cells in sector
   Cell_Count (*)     Array of the number of aircraft in each cell in
                         the sector;
DEFINE TABLES
   UNIT_D             Table for unit of cells or blocks
      unit_id            Cell or block identifier
      aircraft_count     Number of aircraft in the unit;
   SELECT FIELDS aircraft_count
      FROM UNIT_D
      INTO Cell_Count  #local array#
      RETURN COUNT (Tot_No_Cells)
      ORDER BY aircraft_count; #decreasing order#
   Per_Cell_Inv  = 10;  #Per_Cell is 10%#
   CALL Compute_Percent_Of_Aircraft (Per_Cell_Inv IN, Per_Ac OUT,
      Cell_Count IN, Tot_No_Cells IN, Tot_Ac_Count IN);
   Sum_Per_Ac = Per_Ac;
   Per_Cell_Inv = 4;  #Per_Cell is 25%#
   CALL Compute_Percent_Of_Aircraft (Per_Cell_Inv  IN, Per_Ac OUT,
      Cell_Count IN, Tot_No_Cells IN, Tot_Ac_Count IN);
   Sum_Per_Ac = Sum_Per_Ac + Per_Ac;
   Per_Cell_Inv  = 2;  #Per_Cell is 50%#
   CALL Compute_Percent_Of_Aircraft (Per_Cell_Inv IN, Per_Ac OUT,
      Cell_Count IN, Tot_No_Cells IN, Tot_Ac_Count IN);
   Sum_Per_Ac = Sum_Per_Ac + Per_Ac;
END Compute_Unit_Density_Sum;
```

FIGURE 4-17
COMPUTE_UNIT_DENSITY_SUM

```
ROUTINE Compute_Percent_Of_Aircraft;
PARAMETERS Per_Cell_Inv IN, Per_Ac OUT, Cell_Count IN, Tot_No_Cells
   IN, Tot_Ac_Count IN;
DEFINE VARIABLES
   Per_Cell_Inv        Inverse of Per_Cell (proportions of sector's
                          cells)
   Per_Ac              Percent of aircraft corresponding to Per_Cell
   Ac_Count            Number of aircraft contained in Per_Cell
   Tot_Ac_Count        Number of aircraft in the sector
   No_Cells            Number of cells in sector to be processed
   Tot_No_Cells        Total number of cells in sector
   Remainder           Remainder from Tot_No_Cells divided by Per_
                          Cell_Inv
   Cell_Count (*)      Array of the number of aircraft in each cell in
                          the sector
   J                   Index;
   Ac_Count = 0;
   Remainder = MOD(Tot_No_Cells, Per_Cell_Inv);  #MOD is a builtin#
      #function#
   No_Cells = CEIL(Tot_No_Cells/Per_Cell_Inv);  #CEIL is a builtin#
      #function#
   FOR J = 1 TO No_Cells;
      IF J EQ No_Cells
      THEN
         Ac_Count = Ac_Count + Cell_Count(J) * (Remainder/Per_Cell_
            Inv );
      ELSE
         Ac_Count = Ac_Count + Cell_Count (J);
   Per_Ac = Ac_Count/Tot_Ac_Count;
END Compute_Percent_Of_Aircraft;
```

FIGURE 4-18
COMPUTE_PERCENT_OF_AIRCRAFT

and block level respectively. The density measure considers both the cell level and the block level from two perspectives to produce one density value for the sector and time-intervals. The density measure is a number which has a minimum value of 0.0 when the aircraft population is uniformly distribute or scattered over a sector's cells and maximum value of 1.0 when all traffic is concentrated in a single cell.

Figure 4-19 is provided to illustrate the density calculations of the Calculate Basic Sector Workload element and of the Compute Basic Sector Density element. Uniform scattering, moderate clustering, and extreme clustering cases are shown for a given sector and time-interval. The cell aircraft occupancies accumulated over the divisions of a time-interval are presented. The adjacent graphs show the cumulative percent of aircraft occupying the cells starting with the most dense cells. The corresponding values of the percent of the sector's occupancies are also indicated.

The density measure combines the three values of the percent of the sector's occupancy for both the cell and block level. The average of these three values is normalized. The result for the cells is weighted by the Cell Density Ratio and that for the blocks is weighted by 1.0-Cell Density Ratio. The weighted sum (ranging between 0.0 and 1.0) is the value of the density measure. In summary, for the Calculate Basic Sector Workload element, density is calculated as the following:

> Cell_Density_Ratio*((Cell_Density_Value/3-0.283)/(1-0.283))
> + (1-Cell_Density_Ratio)*((Block_Density_Value/3-0.283)/
> (1-0.283)

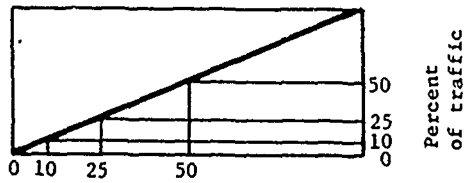
The constant 0.283 is the average percent of the sector's occupancy in the uniform case (i.e., 0.283 = (0.10 + 0.25 + 0.50)/3). The average percent of the sector's occupancy on the cell level is equal to Cell_Density_Value/divided by 3.

The Cell Density Ratio is a parameter that varies from center to center. For centers with small, densely-populated sectors (such as those covering the northeast US), density is better measured using the cell level. The blocks are few in number and may tend to smooth out small-scale traffic clustering patterns significant to the density workload measure. For these centers, the value of Cell Density Ratio should be set close to 1.0. For centers with large, sparsely-populated sectors (such as those covering the western mountains), density is better measured using the block level. In such centers, the cells are numerous, with typical populations of 0 or 1 aircraft. These

Grid-Occupancy Pattern
(Number of aircraft per cell)

2	2	2	2	2
2	2	2	2	2

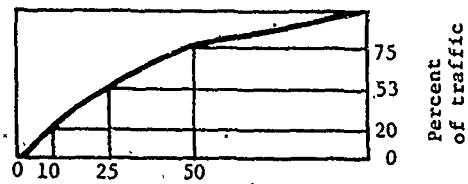
Uniform Scattering
20 aircraft, 10 cells



Percent of cells
DENSITY = 0.00

1	1	3	4	1
1	1	4	3	1

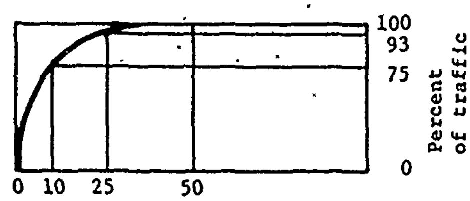
Moderate Clustering
20 aircraft, 10 cells



Percent of cells
DENSITY = 0.29

0	0	3	0	0
0	1	15	1	0

Extreme Clustering
20 aircraft, 10 cells



Percent of cells
DENSITY = 0.91

Three types of clustering are illustrated on the left: uniform scattering, moderate clustering, and extreme clustering.. The adjacent graphs show the cumulative percent of traffic occupying the cells starting with the most dense cells. The density value is listed which is based on the values of percent of cells and percent of traffic indicated on the graph.

FIGURE 4-19
TYPES OF CLUSTERING FOR THE DENSITY MEASURE

cells may be unable to reflect large-scale traffic clustering patterns significant to the density workload measure. For these centers, the value of Cell Density Ratio should be set close to 0.0.

Compute Basic Sector Density

In the Compute Basic Sector Density element, the cell density value and the block density value are computed for each time-interval being processed. The logic for computing the cell density value and the block density value are similar; therefore, only the logic for computing the cell density value is described below. For one time-interval, the records in CELL_DENSITY are examined for each time division of the time-interval and for those records in the basic sector. The aircraft count for each record (cell) is incremented in a local table UNIT_D which contains the aircraft count for each cell. The total number of occupancies in a sector is also accumulated. This number is generally greater than the number of aircraft in a sector.

Compute Unit Density Sum

After all these cells have been processed for the time-interval, the Compute Unit Density Sum element is called. First, the cell occupancy counts in UNIT_D are arranged from the highest to the lowest in a local array (Cell_Count). The most dense 10%, 25%, and 50% of the cells are used to compute corresponding values of the percent of the sector's occupancy. The values of the percent of the sector's occupancies represent various degrees of density from uniform scattering to extreme clustering. In the most uniform case, for values of the 10%, 25%, and 50% of the cells, the corresponding values of the percent of the sector's occupancies are 10%, 25%, and 50%, respectively. In the most extreme case, where all occupancies are in a single cell, the values of the percent of sector's occupancies will be 100%, 100%, and 100%, respectively.

Compute Percent of Aircraft

The percent of the sector's occupancy is computed by the Compute Percent Of Aircraft element and added to the sum of the percent of sector's occupancies (Cell_Density_Value). The Compute Percent Of Aircraft element uses linear interpolation, if necessary, to compute the percent of sector's occupancies.

Calculate Area Workload

If the workload of an area is to be evaluated, then the logic in Calculate Area Workload shown in Figure 4-20 is invoked by the Workload Evaluation component. Under each sectorization plan in effect over the display time intervals, the workload for each basic sector is combined by time-intervals and added to the workload for the respective combined sectors in the COMBINED_SECTOR_WORKLOAD_MEASURES (CSWM) Table. If the supervisor specifies to display the density measure or the overall workload measure, then the Compute Basic Sector Density element (Figure 4-16) is called. The logic of this element is described above in this section.

Compute Combined Sector Calculations

After all basic sectors have been processed for a time-interval, the Compute Combined Sector Calculations is invoked. Figure 4-21 depicts the processing of the Compute Combined Sector Calculations element. Each combined sector in the CSWM Table for the given time-interval is processed. The options for the workload measures of average aircraft count, weighted planned actions, density, and overall workload are tested to determine if the workload values of these measures are to be displayed. Note in Figure 4-21 the similarity in the density equation for a combined sector to that for a basic sector given in Figure 4-15. The difference is that for combined sectors the average percent of the sector's occupancy of cells is equal to the sum of the percent of the sector's occupancy divided by the product of 3 and the number of basic sectors in this combined sector, instead of division by 3 only for a basic sector. After the values of the desired workload measures are computed, they are updated in the CSWM Table.

4.2.2 Threshold Request Component

4.2.2.1 Mission

The supervisor can input a Conditional Mode Request--that is, request to be notified when a workload measure for a given sector has crossed a threshold. This is the purpose of the Threshold Request component. The supervisor enters the sector identification, the workload measure identification and the threshold value.

```
ROUTINE Calculate_Area_Workload;
REFER TO GLOBAL BASIC_SECTOR_WORKLOAD_MEASURES IN, SECTORIZATION_PLAN
    IN, SECTORIZATION_SCHEDULE IN, Display_Time_Horizon IN, COMBINED_
    SECTOR_WORKLOAD_MEASURES OUT;
REFER TO SHARED LOCAL Area_Name IN, Ac_Count_Option IN, Weighted_Pa_
    Option IN, Density_Option IN, Overall_Workload_Option IN, Display_
    Time_Intervals IN;
DEFINE VARIABLES
    Time_Interval_Id  Time interval being processed
    Cell_Density_Value      sum of percent of aircraft for cell
                         density for sector time interval
    Block_Density_Value  Sum of percent aircraft for block
                         density for sector time interval
    Time_Interval     Index
    No_Time_Intervals Number of time intervals to be output
                         for Sectorization_Schedule, Display_ Time
                         Horizon and requested intervals;
#                                                                  #
# In this routine, for purposes of clarity and simplicity, the #
    #following abbreviation is used: #
        #BSWM for BASIC_SECTOR_WORKLOAD_MEASURES#
```

FIGURE 4-20
CALCULATE_AREA_WORKLOAD

```
REPEAT FOR EACH SECTORIZATION_SCHEDULE RECORD
   WHERE SECTORIZATION_SCHEDULE.area EQ Area_Name AND SECTORIZATION_
      SCHEDULE.time is associated with the Display_Time_Horizon and
      the Display_Time_Intervals;
   Count number of time intervals to be output for this
      SECTORIZATION_SCHEDULE record and assign to No_Time_Intervals;
   FOR Time_Interval = 1 TO No_Time_Intervals;
      Compute Time_Interval_Id associated with the Time_Interval;
      REPEAT FOR EACH SECTORIZATION_PLAN RECORD
      #Step through basic sectors in this area under this plan#
         WHERE SECTORIZATION_PLAN.plan_type EQ SECTORIZATION_SCHEDULE.
            plan_type AND SECTORIZATION_PLAN.area_name EQ
            SECTORIZATION_SCHEDULE.area_name;
         INSERT INTO COMBINED_SECTOR_WORKLOAD_MEASURES (sector_number
            = SECTORIZATION_PLAN.combined_sector_number, time_
            interval_id = Time_ Interval_Id, all other fields assigned
            value of 0);
         #Pick out info for this sector during this interval#
         REPEAT FOR EACH BASIC_SECTOR_WORKLOAD_MEASURES (BSWM) RECORD
            WHERE BSWM.sector_number EQ SECTORIZATION_PLAN.basic_
               sector_number AND BSWM.time_interval_id EQ Time_
               Interval_Id;
            IF (Density_Option EQ TRUE) OR (Overall_Workload_Option EQ
               TRUE)
            THEN
               CALL Compute_Basic_Sector_Density (Cell_Density_Value
                  OUT, Block_Density_Value OUT, SECTORIZATION_PLAN.
                  basic_sector_number IN);
            ELSE
               Cell_Density_Value = 0; Block_Density_Value = 0;
            UPDATE IN COMBINED_SECTOR_WORKLOAD_MEASURES (CSWM)(total_
               fl_time = total_fl_time + BSWM.total_fl_time, fp_
               conflict_count = fp_conflict_count + BSWM.fp_conflict_
               count, airspace_conflict_count = airspace_conflict_
               count + BSWM.airspace_conflict_count, pa_counts = pa_
               counts + BSWM.pa_counts, cell_density_value = cell_
               density_value + Cell_Density_Value, block_density_value
               = block_density_value + Block_Density_Value, sector_
               count = sector_count + 1)
               WHERE CSWM.sector_number EQ SECTORIZATION_PLAN.
                  combined_sector_number AND CSWM.time_interval_id EQ
                  Time_Interval_Id;
      CALL Compute_Combined_Sector_Calculations (Time_Interval_Id IN);
END Calculate_Area_Workload;
```

FIGURE 4-20
CALCULATE_AREA_WORKLOAD (Concluded)

```
ROUTINE Compute_Combined_Sector_Calculations;
PARAMETERS Time_Interval_Id IN;
REFER TO GLOBAL Pa_Coefficients IN, Density_Coefficient IN, Airspace_
   Cfl_Coefficient IN, Flight_Plan_Cfl_Coefficient IN, Ac_Coefficient
   IN, Time_Interval_Duration IN, Cell_Density_Ratio IN, COMBINED
   SECTOR_WORKLOAD_MEASURES INOUT;
REFER TO SHARED LOCAL Weighted_Pa_Option IN, Density_Option IN,
   Overall_Workload_Option IN, Ac_Count_Option IN;
DEFINE VARIABLES
   Wpa                        Value of weighted planned actions for
                                 member sector and time interval being
                                 processed
   Den                        Value of density for member sector and
                                 time interval being processed
   Overall                    Value of overall workload for member
                                 sector and time interval being processed
   Ac_Count                   Number of aircraft for member sector
                                 and time interval being processed
   Time_Interval_Id           Identifier of time interval being
                                 processed;
   #                                                                  #
   # In this routine, for purposes of clarity and simplicity, the #
      # following abbreviation is used: #
         #CSWM for COMBINED_SECTOR_WORKLOAD_MEASURES#
```

FIGURE 4-21
COMPUTE_COMBINED_SECTOR_CALCULATIONS

```
REPEAT FOR EACH COMBINED_SECTOR_WORKLOAD_MEASURES (CSWM) RECORD
   WHERE CSWM.time_interval_id EQ Time_Interval_Id;
   Wpa = 0;
   Den = 0;
   Overall = 0;
   Ac_Count = 0;
   IF (Ac_Count_Option EQ TRUE) OR (Overall_Workload_Option EQ TRUE)
   THEN
      Ac_Count = COMBINED_SECTOR_WORKLOAD_MEASURES.total_fl_time/
         Time_Interval_Duration;
   IF (Weighted_Pa_Option EQ TRUE) OR (Overall_Workload_Option EQ
      TRUE)
   THEN
      Wpa = SUM (COMBINED_SECTOR_WORKLOAD_MEASURES.pa_counts *
         Pa_Coefficients);
   IF (Density_Option EQ TRUE) OR (Overall_Workload_Option EQ TRUE)
   THEN
      Den = Cell_Density_Ratio * ((CSWM.cell_density_value/(3 * CSWM.
      sector_count)) - 0.283)/(1 - 0.283) + (1 - Cell_Density_Ratio)
      * ((CSWM.block_density_value/(3 * CSWM.sector_count)) -
      0.283)/(1 - 0.283);
   IF Overall_Workload_Option EQ TRUE;
   THEN
      Overall = Ac_Count * Ac_Coefficient + CSWM.fp_conflict_count
         * Flight_Plan_Cfl_Coefficient + CSWM.airspace_conflict_
         count * Airspace_Cfl_Coefficient + Wpa + Den * Density_
         Coefficient;
   UPDATE IN COMBINED_SECTOR_WORKLOAD_MEASURES
      #in same record under repeat#
      (overall_workload_measure = Overall, aver_aircraft_count = Ac_
      Count, density = Den, weighted_pa = Wpa);
END Compute_Combined_Sector_Calculations;
```

FIGURE 4-21

COMPUTE_COMBINED_SECTOR_CALCULATIONS (Concluded)

4.2.2.2 Design Considerations and Component Environment

Input

The input data to this component are the following:

- Frequency-of-testing parameter (Periodic_Frequency)
- Sector(s), workload measure(s) and threshold(s) to test (WORKLOAD_THRESHOLDS)

The frequency-of-testing parameter is used to determine the frequency for testing of the workload measure(s) against the threshold value(s).

Output

The output requested for the specified sector(s) is displayed with the workload measure(s) and time-interval(s). The output is identified with respect to the threshold which was crossed and to the fact that it is a response to a Conditional Mode Request.

Error Conditions

The logic checks to determine if another Conditional Mode Request already exists for the same sector and workload measure. The supervisor is notified of this situation, if it exists. The supervisor then designates whether to keep the old request and omit the new one or whether to delete the old request and submit the new one. An old request cannot be automatically overridden without the approval of the supervisor.

4.2.2.3 Component Design Logic

The frequency-of-testing parameter specifies at what times the workload output is computed to test against the active thresholds. At these times, the Workload Evaluation component is activated for each sector having one or more thresholds to test. The outputs are stored and then compared with the threshold values. If the threshold has been reached, then the display message and output of the Threshold Request component is presented to the supervisor.

4.2.3 Display Features Component

4.2.3.1 Mission

The Display Features component permits the supervisor to define his own mode of operation and to utilize SWP as he desires. Some possible options are presented in Figure 4-22. The display features provide for automatic outputs, periodic output, establishment of thresholds, and options on the output tables.

4.2.3.2 Design Considerations and Component Environment

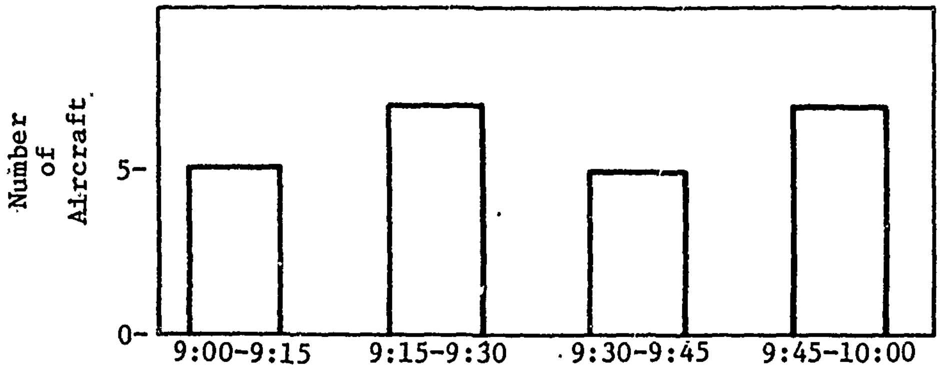
Periodically, an automatic Immediate Mode output of an area or a sector may be requested to activate the Workload Evaluation component. The supervisor may specify a time interval, such as five minutes, as the frequency with which the Workload Evaluation component is invoked and the output data are displayed. (Note the contrast with Conditional Mode, which also causes automatic periodic calls to this component; the periodicity in this case is a parameter, not a supervisor input.)

Various options on the output tables can be defined. Some of these options are as follows:

- To sequence the display of sector data in the area
- To identify the sectorization plans
- To define the display time horizon
- To display only normalized outputs or to include the historical and unnormalized outputs as well
- To select the workload measures to display in the output table
- To select the output table format

The options have default values. In addition to the overall output values, these workload measures may be displayed:

- Average number of aircraft at one time
- Weighted sum of planned actions
- Number of encounters (FPCP and AP)
- Density value



TIME INTERVALS
WORKLOAD MEASURE: AVERAGE NUMBER OF AIRCRAFT
10 MAY 1983 9:02 p.m.

a) Graph for One Sector and One Workload Measure

TIME INTERVAL	AVERAGE NUMBER OF AIRCRAFT
9:00-9:15	5
9:15-9:30	7
9:30-9:45	5
9:45-10:00	7

SECTOR: ENSUE

WORKLOAD MEASURE: AVERAGE NUMBER OF AIRCRAFT
10 MAY 1983 9:02 p.m.

b) Table for One Sector and One Workload Measure

FIGURE 4-22
EXAMPLE OF INFORMATION THAT COULD BE DISPLAYED

WORKLOAD MEASURES	TIME INTERVALS			
	9:00-9:15	9:15-9:30	9:30-9:45	9:45-10:00
Average Number of Aircraft	5	7	5	7
Number of FPCP Encounters	0	1	0	2
Overall	312	419	330	399

SECTOR: ENSUE
10 MAY 1983 9:02 p.m.

c) Table for One Sector and Multiple Workload Measures

SECTORS	TIME INTERVALS			
	9:00-9:15	9:15-9:30	9:30-9:45	9:45-10:00
ENSUE	5	7	5	7
WESTMINSTER	7	9	6	7
SWANN	6	8	7	7
HARRISBURG	4	7	7	8
EAST TEXAS	7	8	6	7

AREA: B
WORKLOAD MEASURES: AVERAGE NUMBER OF AIRCRAFT
10 MAY 1983 9:02 p.m.

d) Table for Multiple Sectors and One Workload Measure

FIGURE 4-22
EXAMPLE OF INFORMATION THAT COULD BE DISPLAYED
(CONTINUED)

SECTORS	TIME INTERVALS			
	9:00-9:15	9:15-9:30	9:30-9:45	9:45-10:00
ENSUE	5 │ 0 │	7 │ 1 │	5 │ 0 │	7 │ 2 │
WESTMINSTER	7 │ 0 │	9 │ 2 │	6 │ 1 │	7 │ 0 │
SWANN	6 │ 0 │	8 │ 1 │	7 │ 1 │	7 │ 0 │
HARRISBURG	4 │ 0 │	7 │ 0 │	7 │ 1 │	8 │ 3 │
EAST TEXAS	7 │ 0 │	8 │ 0 │	6 │ 0 │	7 │ 1 │

AREA: B
WORKLOAD MEASURES: a,b
10 MAY 1983 9:02 p.m.

Key

a	b	c
d	e	f

a. Average Number of Aircraft
b. FPCP Encounter Count
c. AP Encounter Count
d. Weighted Planned Actions
e. Density
f. Overall Measure

e) Table for Multiple Sectors and Multiple Workload Measures

FIGURE 4-22
EXAMPLE OF INFORMATION THAT COULD BE DISPLAYED
(CONCLUDED)

The user-system interface issues play a major role in the development of the display features which have been presented here in general terms.

APPENDIX A

SECTOR WORKLOAD PROBE DATA

This appendix supports the SWP algorithm specification. The data local to SWP are described using the relational data model presented in the "Data Specifications for AERA 1.0" [Vol. 5]. The following data are described in this appendix:

- INDIVIDUAL_AIRCRAFT_WORKLOAD
- REVISED_SUBJECT_WORKLOAD
- ESC_INCREMENT
- ESC_DECREMENT
- CELL_DENSITY
- BLOCK_DENSITY
- SUBJECT_CELL
- SUBJECT_BLOCK
- Parameters

INDIVIDUAL_AIRCRAFT_WORKLOAD:

```
+-------------------------------------------------------------------
| FL_ID | SECTOR_NUMBER | TIME_INTERVAL_ID  | beginning_time
+-------------------------------------------------------------------
     ---------------------------------------------------------------
       | ending_time  | total_fl_time | FPCP_encounter_count
     ---------------------------------------------------------------
     ---------------------------------------------------------
       | AP_encounter_count | altitude_change_pa_count
     ---------------------------------------------------------
     ------------------------------------------------------
       | altitude_change_with_restrictions_pa_count
     ------------------------------------------------------
     ------------------------------------------------------------------+
      | vector_pa_count | speed_change_pa_count | hold_pa_count |
     ------------------------------------------------------------------+
```

pa_counts AGGREGATE (altitude_change_pa_count, altitude_change_with_restrictions_pa_count, vector_pa_count, speed_change_pa_count, hold_pa_count)

This table defines workload associated with a flight ID for a specific sector and time-interval. This table contains a record for each sector and time-interval combination for the flight ID. An IAW Table exists for each flight plan which has been processed.

FL_ID Unique flight identifier.

SECTOR_NUMBER	Basic sector through which the flight passes in the time-interval and for which the values in this record are computed.
TIME_INTERVAL_ID	Time-interval for which the values in this record are computed.
beginning_time	Time when sector-time-interval begins.
ending_time	Time when sector-time-interval ends.
total_fl_time	Total flight time of all the aircraft within the sector during the time-interval specified.
FPCP_encounter_count	Number of Flight Plan Conflict Probe encounters for the flight in this sector-time-interval.
AP_encounter_count	Number of Airspace Probe encounters for the flight in this sector-time-interval.
altitude_change_ pa_count	Number of planned actions of the type--change altitude--for the flight in this sector-time-interval.
altitude_change_with_ restrictions_pa_count	Number of planned actions of the type--change altitude with restrictions--for the flight in this sector-time-interval.
vector_pa_count	Number of planned actions of the type--vector--for the flight and sector-time-interval.
speed_change_ pa_count	Number of planned actions of the type--speed change--for the flight and the sector-time-interval.
hold_pa_count	Number of planned actions of the type--hold on route--for the flight and the sector-time-interval.

pa_counts AGGREGATE (altitude_change_pa_count, altitude_change_with_restrictions_pa_count, vector_pa_count, speed_change_pa_count, hold_pa_count)

REVISED_SUBJECT_WORKLOAD:

```
+----------------------------------------------------------------------
| SECTOR_NUMBER | TIME_INTERVAL_ID | beginning_time | ending_time
+----------------------------------------------------------------------

        ------------------------------------------
          | total_fl_time | FPCP_encounter_count
        ------------------------------------------

        ---------------------------------------------------
          | AP_encounter_count | altitude_change_pa_count
        ---------------------------------------------------

        ------------------------------------------------
          | altitude_change_with_restrictions_pa_count
        ------------------------------------------------

        -------------------------------------------------------------+
         | vector_pa_count | speed_change_pa_count | hold_pa_count |
        -------------------------------------------------------------+
```

pa_counts AGGREGATE (altitude_change_pa_count, altitude_change_with_restrictions_pa_count, vector_pa_count, speed_change_pa_count, hold_pa_count)

This table defines workload associated with a flight ID for a specific sector and time-interval. This table contains a record for each sector and time-interval combination for the flight ID currently being processed in Sector Workload Probe.

SECTOR_NUMBER	Basic sector through which the flight passes in the time-interval and for which the values in this record are computed.
TIME_INTERVAL_ID	Time-interval for which the values in this record are computed.
beginning_time	Time when sector-time-interval ends.
ending_time	Time when sector-time-interval begins.
total_fl_time	Total flight time of all the aircraft within the sector during the time-interval specified.
FPCP_encounter_count	Number of Flight Plan Conflict Probe encounters for the flight in this sector-time-interval.
AP_encounter_count	Number of Airspace Probe encounters for the flight in this sector-time-interval.

altitude_change_pa_count	Number of planned actions of the type--change altitude--for the flight in this sector-time-interval.
altitude_change_with_restrictions_pa_count	Number of planned actions of the type--change altitude with restrictions--for the flight in this sector-time-interval.
vector_pa_count	Number of planned actions of the type--vector--for the flight and sector-time-interval.
speed_change_pa_count	Number of planned actions of the type--speed change--for the flight and the sector-time-interval.
hold_pa_count	Number of planned actions of the type--hold on route--for the flight and the sector-time-interval.

pa_counts AGGREGATE (altitude_change_pa_count, altitude_change_with_restrictions_pa_count, vector_pa_count, speed_change_pa_count, hold_pa_count)

ESC_INCREMENT:

```
+------------------------------------------+
| FL_ID | sector_number | time_interval_id |
+------------------------------------------+
```

This table describes where the subject aircraft will have an encounter. One record is located for each object aircraft having an encounter with the subject.

FL_ID	Unique flight identifier.
sector_number	Basic sector through which the object passes in the time-interval.
time_interval_id	Time interval in which the encounter workload is considered.

ESC_DECREMENT:

FL_ID	sector_number	time_interval_id

This table describes where the subject would have had an encounter. One record is found for each object aircraft which had an encounter with the subject.

FL_ID	Unique flight identifier.
sector_number	Basic sector through which the object passes in the time-interval.
time_interval_id	Time-interval in which the encounter workload is considered.

CELL_DENSITY:

CELL_ID	BEG_TIME	aircraft_count	block_id	sector_number

This table contains density information on the four dimensional (x, y, z, t) airspace cell. The cell is identified in the x, y, z dimensions by CELL_ID and in the t dimension by BEG_TIME. The density information consists of the number of aircraft in the cell and identifying characteristics.

CELL_ID	Unique identifier of an airspace cell in the x, y, z airspace grid.
BEG_TIME	Beginning time of the four dimensional airspace cell.
aircraft_count	Number of aircraft in the four dimensional cell.
block_id	Unique identifier of an airspace block in the x, y, z airspace grid. A block is composed of eight airspace cells.
sector_number	Sector number of the basic sector associated with this cell.

BLOCK_DENSITY:

```
+-------------------------------------------------------+
| BLOCK_ID | BEG_TIME | aircraft_count | sector_number |
+-------------------------------------------------------+
```

This table contains density information on the four-dimensional (x, y, z, t) airspace block. The block is identified in the x, y, z dimensions by BLOCK_ID and in the t dimensions by BEG_TIME. The density information consists of the number of aircraft in the block and the corresponding basic sector.

BLOCK_ID	Unique identifier of an airspace block in the x, y, z airspace grid.
BEG_TIME	Beginning time of the four-dimensional airspace block.
aircraft_count	Number of aircraft in the four-dimensional block.
sector_number	Sector number of the basic sector associated with this block.

SUBJECT_CELL:

FL_ID	CELL_ID	BEG_TIME	aircraft_count	block_id

This table contains density information on the four-dimensional (x,y,z,t) airspace cell by subject aircraft FL_ID. The cell is identified in the x, y, z dimensions by CELL_ID and in the t dimension by BEG_TIME. The density information consists of the number of aircraft in the cell and identifying characteristics.

FL_ID	Unique flight identifier.
CELL_ID	Unique identifier of an airspace cell in the x, y, z airspace grid.
BEG_TIME	Beginning time of the four-dimensional airspace cell.
aircraft_count	Number of aircraft in the four-dimensional cell.
block_id	Unique identifier of an airspace block in the x, y, z airspace grid. A block is composed of eight airspace cells.

SUBJECT_BLOCK:

FL_ID	BLOCK_ID	BEG_TIME	aircraft_count

This table contains density information on the four-dimensional (x,y,z,t) airspace block by the subject aircraft FL_ID. The block is identified in the x, y, z dimensions by BLOCK_ID and in the t dimensions by BEG_TIME. The density information consists of the number of aircraft in the block and the corresponding basic sector.

FL_ID	Unique flight identifier.
BLOCK_ID	Unique identifier of an airspace block.in the x, y, z airspace grid.
BEG_TIME	Beginning time of the four-dimensional airspace cell.
aircraft_count	Number of aircraft in the four-dimensional cell.

Parameters:

Sflid	Flight id of the subject aircraft which initiated SWP.
Trajectory_Update_Status	The update status of the subject aircraft. Has a value of 'NEW' or 'REVISED.'
AP_Not_Dur	The duration of time prior to a AP violation start time that an advisory is displayed to a controller.
Ac_Count_Option	Flag set to TRUE when a display of average aircraft count measure is desired; else set to FALSE.
FPCP_Encounter_Option	Flag set to TRUE when a display of FPCP encounter count measure is desired; else set to FALSE.
AP_Encounter_Option	Flag set to TRUE when a display of AP encounter count measure is desired; else set to FALSE.
Weighted_Pa_Option	Flag set to TRUE when a display of weighted planned actions measure is desired; else set to FALSE.
Density_Option	Flag set to TRUE when a display of density measure is desired; else set to FALSE.
Overall_Workload_Option	Flag set to TRUE when a display of overall workload measure is desired; else set to FALSE.
Delta_T	Size of x,y,t cell in the time dimension.
Area_Name	Name of the area for which workload is being computed.
Organ_Type	Organizational type of "sector" or "area" for which workload is being computed.
Periodic_Frequency	Amount of time between periodic executions of workload evaluation.

Request_Type	Type of workload evaluation desired (single execution or periodic executions).
List_Of_Sectors (*)	Array of sectors for which workload is being computed.
Display_Time_Intervals (*)	Array of time-interval identifiers for which workload is being computed. Time intervals are within the Display Time Horizon.

APPENDIX B

GLOSSARY

The number in parentheses at the end of the definition refers to the section in which the term is first used.

AAS	Advancad Automation System (1.1)
Activation time	The time at which the workload of a planned action is expected to be performed (2.1.8)
Adaptation Factor	A numerical value used to account for the differences in the computed and expected values of the workload measures due to unpredictable features like the quantity and quality of data (2.2.9)
Advisory horizontal separation criterion	The criterion with which Flight Plan Conflict Probe compares distance between aircraft (2.2.5)
AERA	Automated En Route Air Traffic Control (1.4.1.1)
AP	Airspace Probe (1.1)
Area	Second level division (see "Center," "Sector") of the Continental United States airspace. Controllers are specially trained for an area's airspace, a region bounded horizontally by a polygon and stretching vertically from the center floor to 60,000 feet (1.4.1)
Area Manager	The immediate superior of an Area Supervisor (1.4.1)
Area Supervisor	The first-line supervisor of an area (1.4.1)
ARTCC	Air Route Traffic Control Center (see "Center") (1.4.1)
ATC	Air Traffic Control (1.3)

Basic sector	The smallest division of Continental United States airspace (see "Sector," "Area," "Center"). A sector is composed of one or more basic sectors (1.4.1)
Basic Sector Workload Measures Table	A table in the Global Data containing values of workload measures for each basic sector and time interval from the present to the time horizon (3.1.2.1)
Block	A cell in the block density airspace grid (2.1.6)
Block density airspace grid	The four-dimensional airspace grid whose cell size in the x, y, and t dimensions is twice that of the cell density airspace grid, but whose grid size in the z dimension is the same; the grid stores the number of aircraft occupying each cell (2.1.6)
BSWM Table	BASIC_SECTOR_WORKLOAD_MEASURES Table (3.1.2.1)
Cell	Individual parallelepipeds in (x,y,t) space within the airspace grid (2.1.6)
Cell density airspace grid	The four-dimensional airspace grid whose cell size in the x, y, and t dimensions is the same as the sector airspace grid and which stores the number of aircraft occupying each grid cell (2.1.6)
Center	Administrative headquarters and operational facility for control of a first level division (see "Area," "Sector") of the Continental United States airspace (there are currently 20 centers); controls a region bounded horizontally by a polygon and stretching vertically from the center floor to 60,000 feet (1.4.1)
CFC	Central Flow Control (1.4.1.2)
Combined sector	A division of Continental United States airspace (see "Sector," "Area," "Center") which is composed of one or more basic sectors as defined in the sectorization plan (1.4.1)

Combined Sector Workload Measures Table	A table in the Global Data containing values of workload measures for each combined sector and time interval from the present to the time horizon (3.1.2.1)
Component	Third level algorithmic unit in breakdown of AERA (see "Function," "Subfunction," "Element") (1.3)
Conditional Mode	SWP is said to be in Conditional Mode when a supervisor requests that sector workload information be displayed when conditions specified by him are satisfied (2.1.5)
Conditional Mode Request	A request by a supervisor that information be displayed when conditions specified by him are satisfied (2.1.5)
Controller	In this document, an en route radar controller as defined in "Glossary of Common Terms in Air Traffic Control Operations" [6] (1.4.1)
Controller workload	Work performed by controllers at control positions; see also "Sector workload" (2.1.9)
CSWM	COMBINED_SECTOR_WORKLOAD_MEASURES Table (3.1.2.1)
Delta Horizon	The period between consecutive updates of the time horizon (2.1.1)
Display-as-advisory time	The time at which an advisory message is first displayed to a controller (2.2.5)
Element	Fourth level algorithmic unit in breakdown of AERA (see "Function," "Subfunction," "Component") (1.3)
ELOD	En-Route Sector Loading (1.4.1.2)
Encounter	A nominee whose horizontal distance from the subject aircraft is less than a threshold (i.e., it violates the advisory horizontal separation criteria) (2.1.7)

Encounter-status-changed aircraft	Aircraft whose status has changed between encounters and non-encounters, or whose location of encounter or display time has changed (4.1.1.1)
ESC	Encounter-status-changed (4.1.1.1)
ESC Tables	Refers to both ESC Increment Table and ESC Decrement Table whose records contain sector-time-intervals with encounters created or deleted, respectively, due to trajectory updates (4.1.1.2.2)
FAA	Federal Aviation Administration (1.1)
Fix	A specified point on an aircraft's route used for navigation (1.4.1.2)
FPCP	Flight Plan Conflict Probe (1.1)
FPCP Fine Filter	An algorithm that tests subject-nominee segment pairs against FPCP separation criteria using rigorous mathematical analyses (2.2.6)
Function	A major building block of AERA—a principal algorithm which is the top level unit in the breakdown of AERA (see "Subfunction," "Component," "Element") (1.1)
Global Data	The set of data used by more than one AERA function or which is input to or output from AERA (3.1.2.1)
Grid cell	Same as "Cell" (2.1.6)
Grid chain	The set of cells occupied by an aircraft's trajectory (2.1.6)
Holding pattern	An aircraft maneuver to delay its en route progress; usually a circling or spiraling within a specified airspace (2.1.8)
Horizon update	Same as SWP horizon update (2.1.1)
IAW Table	Individual Aircraft Workload Table (4.1.1.1)

Immediate Mode	SWP is said to be in Immediate Mode when a supervisor requests an immediate display of sector workload information (2.1.5)
Immediate Mode Request	A request by a supervisor for the immediate display of sector workload information (2.1.5)
Measure	Same as "Workload measure" (1.4.1)
NAS	National Airspace System (1.1)
NASP	National Airspace System Plan (1.1)
Nominee	In the Flight Plan Conflict Probe, an object aircraft that occupies the same or an adjacent cell as the subject aircraft and whose altitude limits overlap those of the subject (2.1.7)
Object	An aircraft (which is not the subject) whose current trajectory has already been processed by FPCP (2.1.4)
Occupied cell	A cell satisfying certain FPCP or SWP criteria relative to a trajectory (2.1.6)
PDL	Program Design Language (1.2)
Planned action	Any of a set of actions that can be anticipated for an aircraft based on its trajectory (2.1.8)
Radar controller	Same as "Controller" (1.4.1)
Resynchronization	Recomputation of the estimated aircraft trajectory when the trajectory is inconsistent with the aircraft's recent radar track history (1.5.2)
RSW Table	Revised Subject Workload Table (4.1.1.1)
Sector	Third level division (see "Center," "Area," "Basic Sector") of the Continental United States airspace to which a controller is assigned; a region bounded horizontally by a polygon and stretching vertically from the center floor (the ground or a specified altitude) to a ceiling altitude (1.4.1)

Sector Airspace Grid	Grid dividing the horizontal dimensions of the planning region over time into discrete cells (2.1.6)
Sector-time-interval	Portions of a trajectory, the beginning and end of which are at times either when the aircraft enters a new sector or when a new time-interval begins (3.3.1)
Sector workload	Tasks performed in a sector; see also "Controller workload" (2.1.9)
Sectorization	The process of combining or decombining sectors (see "Sector") (1.4.1)
Sectorization plan	A way in which basic sectors can be combined via sectorization; see "Sectorization" (2.1.2)
Sectorization schedule	The current sectorization plan and the sectorization plans and their planned times of implementation through the time horizon (2.1.3)
Subfunction	Second level unit in the breakdown of AERA (see "Function," "Component," "Element") (1.3)
Subject	The aircraft whose new, updated, revised, or alternative (trial probe) trajectory is currently being tested by FPCP (2.1.4)
Subject Aircraft Workload Subfunction	The subfunction of SWP which updates workload data on the subject aircraft (3.0)
Submeasure	The finest subdivision of calculated workload (2.1.9)
Supervisor	Either an Area Supervisor or an Area Manager (1.4.1)
Supervisor Requests Subfunction	The subfunction of SWP that receives and processes certain data entered by the supervisor (3.0)
SWP	Sector Workload Probe (1.1)
SWP display time horizon	The maximum length of time into the future for which SWP values are displayed (2.1.1)

SWP horizon update	An activation of SWP to evaluate workload for new portions of trajectories (2.1.1)
SWP trajectory update	One of four events: 1) a trajectory is added, 2) a trajectory is resynchronized, 3) a trajectory is amended, or 4) a horizon update occurs (2.1.3)
Time Horizon	The length of time into the future for which SWP evaluates its workload measures (2.1.1)
Time-interval	A unit of time for which workload is computed (2.1.1)
Time-interval duration	The length of time in a time interval; the unit in time for computing workload in a sector (2.1.1)
TJE	Trajectory Estimation (1.1)
Trajectory	Description of an aircraft's position in (x,y,z,t) space, produced by applying altitude and timing assumptions to the filed flight plan and revised when necessary (1.5.2)
Trajectory update	Same as "SWP trajectory update" (2.1.3)
Vector	Controller-directed maneuver to provide an aircraft with a change in route (2.1.8)
Workload	Any task performed by personnel to provide air traffic control services to aircraft (1.4)
Workload measure	A mathematical function that assigns a numerical value to one or more aspects of workload (1.4.1)

APPENDIX C

AERA PDL LANGUAGE REFERENCE SUMMARY

C.1 Overview of the Use of AERA PDL

The AERA Program Design Language (PDL) has been created for the single purpose of presenting algorithms in this specification document. It evolves from previous AERA uses, and from MITRE WP-81W552, "All About E," October 1981.

The description of this appendix is intended to support readers and users of AERA PDL. AERA PDL supports readable, yet structured and consistent, descriptions of algorithms.

AERA PDL Features

- Relational data tables can be defined and manipulated by constructs in the language.
- Builtin functions are used to provide routine calculations without showing all of the detail.
- Routines are used to modularize logic paths and data scope.
- Indentation is used to indicate statement grouping, statement continuation, and levels of nesting.
- Routines explicitly define data or refer to predefined data.

AERA PDL Statements

The types of statements used in AERA PDL are:

- English language statements
- assignment statements
- routine declaration statements
- data manipulation statements
- flow of control statements

C.2 Elements of AERA PDL

Keywords

Keywords are words reserved for the usage of AERA PDL. Figure C-1 presents all the keywords used in the current version of AERA PDL, grouped for convenience.

routine construction keywords

CALL	END	ROUTINE

data reference keywords

PARAMETERS	IN
REFER TO GLOBAL	OUT
REFER TO SHARED LOCAL	INOUT
DEFINED IN GLOSSARY	

data definition keywords

DEFINE CONSTANT(S)
DEFINE VARIABLE(S)
DEFINE TABLE(S)

common arithmetic builtin function keywords

AVG	MIN	ABS	EXP	COS	ARCCOS
SUM	MAX	CEIL	LOG	SIN	ARCSIN
PROD	MEDIAN	FLOOR	SQRT	TAN	ARCTAN
		SIGNUM			
		MOD			

coordinate geometry builtin function keywords

DIST	DOT	INTERSECTION
MAGNITUDE	CROSS	INTERPOLATE
DIRECTION	LINE	

set builtin function keywords

UNIQUE	COUNT	CONCAT	BOOL

FIGURE C-1
KEYWORD GROUPINGS

set operator keywords

<u>UNION</u> <u>INTERSECT</u>

table manipulation keywords

<u>SELECT FIELDS</u>	<u>ALL</u>
<u>INSERT INTO</u>	<u>FROM</u>
<u>DELETE FROM</u>	<u>INTO</u>
<u>UPDATE IN</u>	<u>WHERE</u>
	<u>ORDERED BY</u>
	<u>RETURN COUNT</u>

value constant keywords

<u>TRUE</u> <u>FALSE</u> <u>NULL</u>

comparison keywords

<u>NOT</u>	<u>GT</u>	<u>EQ</u>	<u>ANY</u>
<u>OR</u>	<u>GE</u>	<u>NE</u>	<u>ALL</u>
<u>AND</u>	<u>LT</u>	<u>IS IN</u>	
	<u>LE</u>	<u>IS NOT IN</u>	

flow of control keywords

<u>IF</u> ... <u>THEN</u> ... <u>ELSE</u>
<u>CHOOSE CASE</u> ... <u>WHEN</u> ... <u>THEN</u> ... <u>OTHERWISE</u>
<u>FOR</u> ... <u>TO</u>
<u>REPEAT WHILE</u>
<u>REPEAT UNTIL</u>
<u>REPEAT FOR EACH</u> ... <u>RECORD</u>
<u>GO TO</u>

FIGURE C-1 (Concluded)
KEYWORD GROUPINGS

Operators

The operators of AERA PDL are summarized in Figure C-2.

The Assignment Operator

- The format of the assignment statement is:
  "target" = "expression"

- The target may be any type of data allowed by AERA PDL.

- The assignment operator includes the ability to fill a table from data contained in other tables. The form of this use of the assignment operator is:
  "table_name" = "table_expression" ;

Builtin Functions

The builtin functions of AERA PDL accept either an single value or data organized into an array. The author of a routine must make it clear in comments and text what form of data is being processed by the builtin function. Builtin functions are listed in Figure C-3.

C.3 Routine Construction

The order of appearance of constructs in a routine is:

- ROUTINE — required
- PARAMETERS — optional
- REFER TO GLOBAL — optional
- REFER TO SHARED LOCAL — optional
- DEFINED IN GLOSSARY — optional
- DEFINE CONSTANTS — optional
- DEFINE VARIABLES — optional
- DEFINE TABLES — optional
- logic flow — required, but will vary by routine.
- END — required

Three of the constructs are noted below:

The ROUTINE Construct

- The ROUTINE construct names the routine.

- The syntax of the ROUTINE construct is:
  ROUTINE "routine_name" ;

assignment operator

A = B	A is assigned the value of B

arithmetic operators

A + B	A plus B
A - B	A minus B
A * B	A times B
A / B	A divided by B
A ** B	A to the power of B

comparison operators

A LT B	A is less than B
A LE B	A is less than or equal to B
A GT B	A is greater than B
A GE B	A is greater than or equal to B
A EQ B	A is equal to B
A NE B	A is not equal to B

logical operators

NOT A	The logical opposite of A
A OR B	Logical OR of A and B
A AND B	Logical AND of A and B

set operators

A INTERSECT B	The set intersection of A and B
A UNION B	The set union of A and B
A IS IN B	A is an element of the set B
A IS NOT IN B	A is not an element of the set B

FIGURE C-2
GROUPINGS OF AERA PDL OPERATORS

FUNCTION	MEANING
ABS(x)	Absolute value of x
ARCCOS(x,y)	Inverse cosine of the ratio of y to x
ARCSIN(x,y)	Inverse sine of the ratio of y to x
ARCTAN(x,y)	Inverse tangent of the ratio of y to x
AVG(A)	Mean of the elements in A
BOOL(x)	Numerical equivalent of logical condition: 1 if x is TRUE, 0 if x is FALSE
CEIL(x)	Smallest integer greater than or equal to x
CONCAT(s1,s2,...,sN)	Concatenation of strings s1 through sN
COS(x)	Cosine of x
COUNT(A)	Number of elements of a set A
CROSS(v1,v2)	Cross product of vectors v1 and v2
DIRECTION(p1,p2)	Direction of p2 from p1 in degrees from the north; usually will be expressed in degrees clockwise from true north
DIST(p1,p2)	Euclidean distance between points p1 and p2
DOT(v1,v2)	Dot product of vectors v1 and v2
EXP(x)	e to the x power
FLOOR(x)	Greatest integer less than or equal to x

FIGURE C-3
BUILTIN FUNCTIONS

FUNCTION	MEANING
INTERPOLATE(a,b,t)	The point (1-t)a+tb
INTERSECTION(L1,L2)	The point of intersection of the lines L1 and L2
LINE(p1,p2)	Vector (a,b,c) corresponding to the line ax + by = c which passes through the points p1 and p2
LOG(x)	Log of x in base e
MAGNITUDE(v)	Length (i.e., norm) of the vector v
MAX(A)	Largest of the elements in the set A
MEDIAN(A)	Median value of the elements in set A
MIN(A)	Smallest of the values in set A
MOD(x1,x2)	Remainder when x1 is divided by x2
PROD(A)	Product of the elements in A
SIGNUM(x)	Function yielding 1 if x GT 0, -1 if x LT 0, and 0 if x EQ 0
SIN(x)	Sine of x
SQRT(x)	Square root of x
SUM(A)	Sum of the elements in A
TAN(x)	Tangent of x
UNIQUE(A)	The set A with no duplicate elements

FIGURE C-3 (Concluded)
BUILTIN FUNCTIONS

The CALL Construct

- The CALL construct invokes use of another routine as a subroutine and passes to it the data on which it is to operate.

- The syntax of the CALL construct is:
  CALL "routine_name" ("data_usage_list") ;

- The data usage list in the CALL statement must match in number and data utilization (IN, OUT, INOUT) the PARAMETERS statement of the called routine.

The END Construct

- The END construct shows the formal end of the routine.

- The syntax of the END construct is:
  END "routine_name" ;

C.4 Data Definitions

Data usage is defined in the constructs placed at the beginning of each routine.

The structures, or organization of data, recognizable to AERA PDL include constants, atomic variables, hierarchically structured variables, arrays, tables, and field-types. The hierarchically structured variables are the same as the structure variables of PL/I.

Within a table, the values corresponding to the definition of a field-type are called fields when they are referred to individually. The values for a whole column of a table (or a subset of the whole column) may be referred to as a set of fields.

The following data definition constructs appear in the order shown, if any are needed. The first line of each construct begins in column 1, aligned with the ROUTINE construct.

The PARAMETERS Construct

- This construct provides usage information about the data that are being provided by the calling routine in the form of specification of read-only 'IN', write-only 'OUT', or modification of an existing value 'INOUT'.

- Variables appearing in the PARAMETERS construct are still local data for the routine being defined and as such appear in the definition constructs.

- The syntax of the PARAMETERS construct is:
  PARAMETERS "data_usage_list" ;

The REFER TO GLOBAL Construct

- This construct provides reference to, and usage information for, data from the Global data model.

- The syntax of the REFER TO GLOBAL construct is:
  REFER TO GLOBAL "data_usage_list" ;

The REFER TO SHARED LOCAL Construct

- This construct provides reference to, and usage information for, data from the Shared Local data model described in Appendix A of the specification.

- The syntax of the shared local construct is:
  REFER TO SHARED LOCAL "data_usage_list" ;

The DEFINED IN GLOSSARY Construct

- This construct provides reference to, and usage information for, data from a specially prepared Glossary that centralizes the definition of data variables that are used repeatedly within a given function of the algorithmic specification.

- The syntax of the shared local construct is:
  DEFINED IN GLOSSARY "data_usage_list" ;

The DEFINE CONSTANTS Construct

- The use of named constants instead of in-line numerical constants is available at the discretion of the author of an algorithm. Named constants, if present, are to be declared with this construct.

- The syntax of the DEFINE CONSTANTS construct is:
  DEFINE CONSTANTS "constant_definition_block" ;

The DEFINE VARIABLES Construct

- The syntax of the DEFINE VARIABLES construct is:

```
DEFINE VARIABLES "variable_definition_block" ;
```

The DEFINE TABLES Construct

- The syntax of the DEFINE VARIABLES construct is:

```
DEFINE TABLES "table_definition_block";
```

C.5 Flow of Control Constructs

The IF...THEN...ELSE Construct

- The syntax of the IF...THEN...ELSE construct is:

```
IF "condition"
   THEN
      "statement_block"
 [ ELSE
      "statement_block" ]
```

The CHOOSE CASE Construct

- This construct provides a choice of one of several alternative logic paths depending on the first condition satisfied among the conditions specified.
- The OTHERWISE phrase is optional.
- The syntax of the CHOOSE CASE construct is:

```
CHOOSE CASE
   WHEN "condition" THEN
      "statement_block"
 [ WHEN phrases repeated as necessary ]
 [ OTHERWISE
      "statement_block" ]
```

The REPEAT WHILE Construct

- The syntax of the REPEAT WHILE construct is:

```
REPEAT WHILE "condition" ;
   "statement_block"
```

The REPEAT UNTIL construct

- The syntax of the REPEAT UNTIL construct is:

```
REPEAT UNTIL "condition" ;
   "statement_block"
```

The REPEAT FOR EACH RECORD Construct

- This construct explicitly loops over all records in table, or the subset of a table as specified in a WHERE phrase.

- The syntax of the REPEAT FOR EACH construct is:

```
REPEAT FOR EACH "table_name" RECORD
  [ WHERE "condition" ] ;
     "statement_block"
```

- Within the statement block of this loop, the construct of "table_name"."field_name" means only the ONE value that is associated with the record for that iteration of the loop.

- If it is necessary to refer to entire columns of the table that is being looped on, the correct form of the reference is ALL("table_name"."field_name"). This construct means exactly what "table_name"."field_name" would have meant if the loop had not been in effect.

The GO TO Construct

- The syntax of the GO TO construct is:

```
GO TO "label" ;
```

The FOR...TO... Construct

- The syntax of the FOR...TO... construct is:

```
FOR "loop_index" = "initial_value" TO "last_value" ;
    "statement_block"
```

C.6 Table Manipulation Constructs

The SELECT FIELDS Construct

- This construct extracts data from a table, or from a collection of tables, and makes it available to the routine.

- The syntax of the SELECT FIELDS construct is:

```
SELECT FIELDS [ UNIQUE ] [ "field_list" | ALL]
   FROM "table_name_list"
  [ INTO "local_variable_name_list" ]
  [ WHERE "condition" ]
  [ ORDERED BY "field_name" ]
  [ RETURN COUNT ( "local_variable" ) ] ;
```

The INSERT INTO Construct

- This construct allows a new record to be inserted into a table.

- The syntax of the INSERT INTO construct is:
  INSERT INTO "table_name" ("field_assignments")
  [WHERE "condition"] ;

- All insertions will preserve the assumption of no duplicate records allowed in the table.

The UPDATE IN Construct

- This construct allows existing records in a table to have certain of their values changed.

- The syntax of the UPDATE IN construct is:
  UPDATE IN "table_name" ("field_assignments")
  [WHERE "condition"] ;

The DELETE FROM Construct

- This construct removes selected records from a table.

- The syntax of the DELETE FROM construct is:
  DELETE FROM "table_name"
  [WHERE "condition"] ;

C.7 Glossary

"comparison"

- There are four possible syntaxes for the comparison. These are not given separate names, but will all be shown as if they shared the same element of the language.

- The first syntax is for arithmetic comparisons:
  "individual" GE|LE|GT|LT "individual"

- The second syntax is for general comparisons:
  "individual" EQ|NE "individual"

- Both of these syntaxes are also valid if they are used to compare two variables with the same complex organization, for example two vectors of the same length or two field types from the same table. In this case the result has as many answers as there are elements in the compared variables.

- The third syntax is for arithmetic comparisons:
  "individual" GE|LE|GT|LT ANY|ALL "set"

- The fourth syntax is for general comparisons:
  "individual" IS IN|IS NOT IN "set"

- The latter two syntaxes are used to qualify any individual based on any value in a set of values.

"condition"

- The syntax of the condition is:
  "comparison" [AND|AND NOT|OR|OR NOT "comparison"]

- The optional part of this syntax can be repeated as often as required.

"constant definition block"

- The content of the constant definition block is three columns: the constant names, the constant values, and the constant descriptions.

- The constant names are aligned in a column 3 spaces indented from the DEFINE CONSTANTS line.

- The other two columns are aligned as convenient, so that there is no visual overlap between the columns.

"data usage list"

- A routine must declare the type of use for all of its data that are known outside the routine.

- The three types of use are: read only (IN), create (OUT), and modify an existing copy (INOUT).

- The format of a data usage list is:
  "variable_name" "usage_type", ...

- An example of the format for data usage list is:
  An_Input_Parameter IN, A_LOCAL_TABLE INOUT

"expression"

- Variables may be formed implicitly in expressions without being separately named or defined.

- Expressions are combinations of defined variables with the operators and builting functions of AERA PDL.

- In an expression, the implicit variable output from any builtin function may be used as the input to any other builtin function or operator.

- An expression, when fully evaluated, yields one variable.

"field assignments"

- This term only appears in statements referring to exactly one table: INSERT and UPDATE.

- The form of the term is a comma-separated list:
  "field_assignment", ...

- The form of a single assignment is:
  "field_name" = "value_expression"

- In this term the field_names do not have to be qualified by the table name (that is given in the statement).

"table definition block"

- Three types of definition are made in this block: table definitions, field-type definitions, and AGGREGATE definitions.

- Table definition lines are formatted as:
  "table_name" "table_definition"

- Field-type definitions lines are formatted as:
  "field_name" "field_definition"

- Aggregate definitions are formatted as:
  "aggregate_name" AGGREGATE ("field_name_list")

- Fields will contain only atomic (single-valued) variables.

- Aggregates may be used so that a program can manipulate multiple fields in one statement when it makes sense to do so.

"table-expression"

- Tables may be used implicitly in assignments or comparisons being separately named or defined.

- A table expression is either a table name or a SELECT statement specifying the contents of the implicit table.

"table name"

- Generally, this is just the name of a table.

- In a few statements, there is a syntax that allows a user to define a synonym and use it in the rest of that statement. The intent of this option is to allow shorter where clauses that are easier to read. The format of the synonym reference is:
  "existing_table_name" ("synonym")

- The statements that allow this use are those that have the where clause: SELECT, INSERT, DELETE, UPDATE, and REPEAT.

"variable definition block"

- The content of the variable definition block is two columns: variable names and variable descriptions.

- Align variable names in a column that is indented 3 spaces from the DEFINE VARIABLES line.

- Align variable definitions in a column as convenient; when a structure element is defined, both the variable name and the variable definition should be indented three spaces from the name and definition of the next higher level variable.

- Three types of variables may be defined in this block: atomic variables, arrays, and structured variables.

- Each element variable is described by a line:
  "variable_name"   "variable_definition"

- Each array variable is described by a line:
  "variable_name" ("dimensions") "variable_definition"

- Each structured variable is described by multiple lines, one line per lowest level element, and one line for each named level of grouping/structure, with indentation levels used to indicate the grouping.

- The names of subordinate elements of a structured variable are named in all lower case letters.

- The use of complex structured variables is not encouraged; one reasonable use for them is to receive the values of AGGREGATEs.

C.8 Other Uses and Conventions

Use of Special Characters in AERA PDL

- Parentheses are used for grouping statements and setting off special parts of the constructs.
- Semicolons are used as statement terminators.
- Colons are used to terminate labels.
- Underscore is used to separate words in multi-word identifiers.
- The symbols '+','-','*', and '/' are used as arithmetic operators.
- The pound sign '#' is used as a comment delimiter, for beginning and end of each comment line.
- Commas are used as separators in lists of operands.
- Periods are used to separate fully qualified names.

Naming Conventions

- Keyword identifiers use only uppercase letters and are underlined. They are the only underlined identifiers in the PDL.
- Table identifiers from the relational data base also use only uppercase letters.
- AGGREGATE identifiers for combinations of fields use no uppercase letters.
- References to fields in a table, used in the normal course of reference in AERA PDL, will be fully qualified by including the table name.

Other Identifiers

- Identifiers for constants, routines, labels, arrays, and hierarchically structured variables are all be named using word-initial capitals.

- For hierarchically structured variables, all of the subordinate elements within the structure use only lowercase letters.

- For hierarchically structured variables, all references to the subordinate elements in the structure will be in fully qualified form using separate identifiers.

- Global data and shared local data can include both tables and parameters. The individual parameters are named using word-initial capitals.

Use of the Formal Constructs in AERA PDL Statements

- Statements may use formal constructs or clear English descriptions to specify the intended test or action.

- Any AERA PDL statement is terminated by a semicolon, including any English statement outside of a comment.

- Below the level of statement, some statements have a finer organization in terms of "phrases", usually occupying a line per phrase and indented one level from the first line of the original statement.

Statement Organization

- Indentation is used to indicate statement grouping, statement continuation, and levels of nesting.

- Any statement may have a label starting in column 1.

- Continuation lines are indented three spaces from the original line of the statement.

- Comments are used as needed, bracketed by the special character '#'.

APPENDIX D

REFERENCES

1. U.S. Department of Transportation, Federal Aviation Administration, Advanced Automation System: System Level Specification, FAA-ER-130-005B, April 1983.

2. U.S. Department of Transportation, Federal Aviation Administration, National Airspace System Plan: Facilities, Equipment, and Associated Development, April 1983.

3. William J. Swedish, Barbara C. Zimmerman, Audrey W. Lipps, J. Glenn Steinbacher, "Operational and Functional Description of AERA 1.01," MTR-83W69, The MITRE Corporation, McLean, Virginia, September 1983.

4. William J. Swedish, "Evolution of Advanced ATC Automation Functions," WP-83W149, The MITRE Corporation, McLean, Virginia, March 1983.

5. Audrey W. Lipps, William J. Swedish, Barbara C. Zimmerman, "Operational and Functional Description of the AERA Packages," MTR-83W125, The MITRE Corporation, McLean, Virginia, September 1983.

6. Glenn C. Kinney, Glennis L. Bell, and Martin A. Ditmore, "Glossary of Common Terms in Air Traffic Control Operations," WP-83W22, The MITRE Corporation, McLean, Virginia, March 1983.